

折紙の幾何的な制約を考慮した形状設計に関する研究

著者	鶴田 直也
発行年	2015
学位授与大学	筑波大学 (University of Tsukuba)
学位授与年度	2014
報告番号	12102甲第7284号
URL	http://hdl.handle.net/2241/00129420

折紙の幾何的な制約を考慮した 形状設計に関する研究

鶴田 直也

システム情報工学研究科
筑波大学

2015 年 1 月

概要

紙を折ることで形を作り出す「折り紙」は、日本に古くから伝わる遊びであり、我々の多くが幼少期に体験するものである。しかし、単なる遊びとして捉えられるだけでなく、紙を折る操作と幾何学の密接な関係から数学の研究対象としても着目されてきた。ものを折り畳んで小さくすることは、収納や運搬に必要なスペースを減らすという利点があり、近年では、折り紙の折り畳み構造を工学的に応用するための研究も活発になっている。その一方で、紙のような伸縮しない素材の折り曲げのみで制作可能な形は自由度が低く、意図した形状を作り出すことは困難な課題である。

この問題に取り組んだこれまでの種々の研究は、伝統的な折り紙にない緻密さや意匠性を持つ形状を実現し、折り紙の応用可能性を少しずつ示してきた。折り紙の世界は多くの価値ある形状を内包していると考えられるが、未だ開拓されていない領域を残す分野でもある。折り紙の形状設計問題に取り組み、これまで知られてこなかった折り方や形状を発掘することは、折り紙界の発展だけでなく、他分野との融合によって新たな価値を創出する機会を増加させるという大きな意義がある。そして、これを達成するためには、多種多様な設計手法を確立していくことが必要不可欠である。

本研究では、折り紙において最も基本となる平坦に折り畳めるもの、近年注目されている立体的な曲げを含むもの、そしてユニット折り紙と呼ばれる複数枚の紙を部品のように組み合わせるものの3つのテーマに対して、それぞれに新しい設計アプローチおよびシステムを提案する。1つ目は、折り畳み形状提示による平坦折り紙作品の発見手法で、既存の設計手法では困難だった折り回数の少ない作品を対象としている。2つ目はスリット（細長い切り込み）を入れた平板素材の折り曲げ形状を設計するシステムであり、切込みを許容することで自由度の高い設計を可能にした。最後は、ユニット折り紙と呼ばれる、複数の小さな部品を折ってそれらを組み合わせて大きな立体を作る折り紙を対象としたものである。基礎となる構造が正三角形集合で表現されることに着目して、そのような正三角形集合の立体の設計を行っている。これらは、いずれも折り紙の裾野を広げる上で重要なテーマである。

上記のシステムを構築し、出力として作例を示すことで、これまでの手法では設計が困難、あるいは発見が難しい形状が生成できることを確認した。また、これらのシステムを通して得られた結果を元に、折り曲げによる形状設計の難しさと折り紙研究の展望について議論する。

目次

第1章	序論	1
1.1	はじめに	1
1.2	現代の折り紙	2
1.3	本研究の目的	3
1.4	本論文の構成	4
第2章	関連研究	5
2.1	平坦折り可能性	5
2.1.1	局所平坦条件	5
2.1.2	折り畳まれた状態の存在判定	6
2.1.3	Map Folding と Stamp Folding	6
2.2	折り紙設計	7
2.2.1	平坦折りの折り紙設計	7
	TreeMaker	7
	シルエットからの設計	7
	折り紙テッセレーション	8
	剛体折り紙	8
	折り紙建築	8
2.2.2	非平坦折りの折り紙設計	9
	直線折り	9
	曲線折りと可展面	9
2.3	ユニット折り紙	10
2.3.1	多面体の設計	11
第3章	折り畳み形状提示による折り紙作品発見の支援	12
3.1	既存の設計手法の問題点	12
3.2	折り畳み形状の列挙	13
3.2.1	折り畳み形状の生成と重複判定	14
3.2.2	折り畳み形状の列挙結果	14
3.3	ユーザ入力に似た作品の検索システム	16
3.3.1	検索システムのインタフェース	16
3.3.2	形状の類似度評価	17

3.3.3	検索システムの実行結果	18
	紙の表裏を考慮した検索	19
	顔のパーツの表示について	19
3.4	ランダムな折り畳み形状提示	19
3.4.1	システムの概要	21
3.4.2	ランダムな形状生成	22
3.4.3	折りの位置	22
3.4.4	折る方向と折り畳み可能な枚数	24
3.4.5	左右対称な折り畳み形状生成	24
3.4.6	インタフェース	26
3.4.7	システムの実行結果と発見した作品例	27
3.5	考察	29
第4章	スリットを含む平板素材の曲げ形状設計	31
4.1	対象とする素材の形状	32
4.2	最適化による形状の修正	33
4.3	ユーザ入力と作業プロセス	35
4.4	結果と考察	37
4.4.1	測定点群への近似	37
4.4.2	実制作	38
4.5	考察	40
第5章	ユニット折り紙の組み合わせ構造の設計	41
5.1	対話的なデルタ多面体の設計システム	42
5.1.1	多面体の連結	43
5.1.2	引き伸ばし	44
5.1.3	錐形部位の凹凸を反転する	44
5.1.4	面の細分割	45
5.1.5	結果と考察	45
5.2	デルタ多面体のグラフと形状に関する調査	46
5.2.1	デルタ多面体のグラフ	46
5.2.2	グラフからの立体構築手法	47
5.2.3	結果	50
5.2.4	デルタ多面体として構築できないグラフ群の発見	52
5.2.5	考察	53
第6章	結論と展望	56
6.1	結論	56
6.1.1	折り畳み形状提示による作品発見	56

6.1.2	スリットを含む平板素材の曲げ形状設計	56
6.1.3	ユニット折り紙の組み合わせ構造設計	57
6.2	展望	57
	謝辞	58
	参考文献	59
付 録 A	自動生成された折り紙形状	66
付 録 B	3 次 3 連結平面グラフを各面が正三角形になるよう構築した多面体	68

図目次

2.1	鶴の折り図	6
2.2	鶴の展開図	6
2.3	平坦に折り畳める頂点: $\theta_1 + \theta_3 = \theta_2 + \theta_4 = 180^\circ$ となる	6
2.4	園部式ユニットによるくす玉 (左: 正二十面体ベース 30 枚組、右: 正八面体ベース 12 枚組)	11
3.1	幼児向けの折り紙作品の例; 左から「いぬのかお」、「チューリップ (はな と はっぱ)」、「ヨット」(文献 ^[79] より)	13
3.2	1 つの折り線から得られる形状; (a) では 2 枚の紙が重なっており、谷折りで (b) と (c)、山折りで (d) と (e)、折ってひらく操作で (f) が得られる	15
3.3	同一形状の判定に用いる値の算出	15
3.4	列挙した形状から発見した作品の例: 左から、3 回折りのキツネ、4 回折りの ティーポット、4 回折りの鳥	16
3.5	システム概要	17
3.6	システム実行画面 (左: 入力ウィンドウ、中央: 結果表示ウィンドウ、右: 折り 手順表示ウィンドウ)	18
3.7	入力形状と検索結果、および実際に折った作品	20
3.8	紙の表裏を考慮した検索例: (a) 入力; (b) 輪郭のみを用いた検索結果; (c) 表裏を 考慮した検索結果	20
3.9	中割り折りによる色の変更効果	21
3.10	目の有無による認識の違い	21
3.11	提案システムのインタフェース	22
3.12	自由度のない折り操作と自由度のある折り操作: (a) カドをカドに合わせるよう に折る; (b) カドを紙のフチに合わせるように折る	23
3.13	1 自由度の折り操作: (a) 点を通るように折る; (b) 直線に垂直に折る; (c) 点を直 線に乗せるように折る	23
3.14	2 種類の対称軸	24
3.15	折りが対象軸と交差しない場合、全体の折り操作は最大 2 回の折りになる	25
3.16	折りが対象軸と交差しない場合、全体の折り操作は開いてつぶす折りになる	25
3.17	折り図の表示画面	26
3.18	自動生成したマナティの折り図	27
3.19	Paula Versnick 作「2 fold santa ^[73] 」とシステムで自動生成された形状	27

3.20	子供が発見した作品例	28
3.21	ワークショップの様子	29
3.22	左右対称な形状生成によって発見された作品例	29
3.23	オンラインデータベースに登録された作品例 (括弧内の数字は折り回数)	30
4.1	スリットを入れた素材の折り曲げによる椅子とその展開図 (成瀬・猪熊建築設計事務所 ^[52])	32
4.2	椅子の展開図; L_1 から L_4 までの連続値およびストリップの数 N によって定義される	32
4.3	ストリップを分割する折り目と最適化計算後のイメージ	33
4.4	3次元座標の計算	33
4.5	滑らかさの計算	34
4.6	端点の位置拘束	34
4.7	四角形領域の縮退排除	35
4.8	ユーザによる位置指定	35
4.9	提案システムの実行画面	35
4.10	ストリップの細分割: 左から分割前、分割後、最適化後	36
4.11	脚部の形状と対称面上の断面形状の指定	36
4.12	作例: 左 2 つは 6 本と 10 本の椅子のモデル、中央はスキャンした点群に近づけたモデル、右 2 つはモニュメントのような形状	38
4.13	3D スキャンで測定した点群データと編集集中の形を重ねて描画した図	39
4.14	3D プリンタで出力したサポート部材とその表面に貼り合わせた画用紙、形成した塩化ビニル板; サイズ (W/H/D): $247 \times 233 \times 179mm$	39
5.1	凸デルタ多面体 (括弧内の数字は多面体を構成している面数)	42
5.2	凸でないデルタ多面体の例 (左: Boat、中: Stella octangula、右: Stellation of Icosahedron)	42
5.3	システム実行画面	43
5.4	多面体の連結操作	44
5.5	引き伸ばし操作	44
5.6	正二十面体の錐形部位を反転する	45
5.7	正四面体の各面を 4 等分した例	45
5.8	対話的な編集によって作成したモデル例	45
5.9	折り紙ユニット 96 枚を組み合わせて制作したウサギのモデル (左上: 基本ユニット、左下: 3 枚組の正三角形)	46
5.10	多面体グラフとデルタ多面体グラフの包含関係	47
5.11	正三角形集合の多面体; 上 1 つのみデルタ多面体であり、下 2 つはデルタ多面体ではない	48
5.12	9_{50} のグラフから構築できる複数のデルタ多面体	48

5.13	デルタ多面体の構築プロセス	49
5.14	三角形集合のグラフを生成するための 3 操作	51
5.15	7 頂点のグラフから構築された多面体	52
5.16	8 頂点のグラフから構築された多面体	54
5.17	$(8_7, N)$ のグラフと正四面体を連結した $(9_{26}, N)$ のグラフ、および対応する立体	55
5.18	8_7 のグラフを含むデルタ多面体を構築できないグラフ	55
5.19	単一面による連結と複数面による連結	55
A.1	1 回折り (全 4 通り)	66
A.2	2 回折り (全 37 通り)	66
A.3	3 回折り (1,507 通り中の 120 例)	66
A.4	4 回折り (136,284 通り中の 270 例)	67
B.1	9 頂点のグラフから構築した多面体 (50 通り中の 30 例)	68
B.2	10 頂点のグラフから構築した多面体 (233 通り中の 36 例)	69

第1章 序論

1.1 はじめに

紙を折ることで形を作り出す「折り紙」は、日本に古くから伝わる遊びであり、我々の多くが幼少期に体験するものである。しかし、単なる遊びとして捉えられるだけでなく、紙を折る操作と幾何学の密接な関係から数学の研究対象としても着目されてきた。それらの研究成果からは紙の折り曲げによって形状を設計するための、様々な手法が提案されている。さらに近年では、折り紙における折り畳み構造を工学的に応用するための研究も盛んである。

折り紙がこうした広がりを見せる背景には、折り紙設計という概念の普及がある。折り紙設計とは、数学的な理論やアルゴリズムに基づいて紙の折り曲げによってできる形状を設計するというもので、従来の試行錯誤による作品創作では実現できなかった昆虫や恐竜のような複雑な形を創り出すことに成功した。折り紙設計の登場以降、作品の細部への作り込みや緻密さの水準は大きく向上し、また、具象的な作品に向けた設計手法や幾何的な折り畳みパターンを折り出すための手法など様々な設計手法が提案されるようになった。

設計のための技術を数学的な手続きによって表現することは、試行錯誤の手間を軽減するだけでなく、計算機で折り紙を扱うことにもつながった。計算機の性能向上も伴い、紙の折り曲げによって生じる複雑な幾何形状をデータとして取り扱うことが可能になってきたことも工学応用が広がった要因の一つである。計算機上で折り紙を取り扱うことは、膨大な数の折り畳みパターンを列挙し、手作業による試作の手間の軽減する、あるいは折り畳み可能な形状を自動生成し、計算機でしか設計し得ない形を作り出すなど、様々な利点がある。このような計算機による貢献もあり、折り紙の世界からは工学的に利用価値の高い構造や芸術性の高い形状が発見されてきた。

しかしながら、折り紙の世界には未だ未知な領域が残されている。折り紙で設計可能な形状の範囲は限られている一方で、そこには多くの価値ある形状が埋もれていると考えられる。これまでに知られてこなかった形状を発見し折り紙の世界の資産を増やすことは、折り紙界の発展だけでなく、他分野との融合によって新たな価値を創出する機会を増加させるという、大きな意義がある。そして、これを達成するためには、折り紙で実現できる形状を設計するための多様な手法を確立することが必要不可欠である。

そこで本研究では、折り紙の形状設計についての研究を行うものとした。

1.2 現代の折り紙

子供の遊戯として認識されることが多い折り紙であるが、近年の目覚ましい発展により、その見方は変化しつつある。以下では、冒頭で述べた工学分野での応用研究の現状や教育での利用、芸術作品としての評価について紹介する。

折り紙の工学応用

折り紙の折り畳み構造は、ものを収納するためのスペースを節約することにつながる。また、構造を工夫することで素早く滑らかに展開できるという利点もある。このため、収納と展開が重要となる製品への活用が期待できる。

折り紙の技術を利用した研究の代表例としては、人工衛星に載せる太陽電池パネルの設計^[47]が挙げられる。大型のパネルを宇宙で運用する際に、小さく折り畳んだ状態で運搬し宇宙で展開することで、小さな部品に分解してから組み立てるという作業が不要になるという利点がある。自動車のエアバッグ^[12]も普段は折り畳まれており衝突時に展開されるもので、瞬時に開くことが求められる。医療器具のステント^[31]に活用した試みも報告されている。ステントは詰まった血管に挿入し、器具を展開することで血管を広げる目的で使用されるものである。この他にも、災害時に使用する避難用の簡易シェルター^[11]や音響効果を変化させる天井のパネル^[58]などの事例がある。より身近な例としては、家具や衣類が挙げられる。パイプ椅子とは異なる、面で折り曲げて立体にする家具^[18]や、捻るような折り畳み構造をデザインに取り入れた服^[27]が販売されている。

教育素材としての折り紙

教育、知育素材として折り紙に高い価値があることは多くの研究者が着目している。紙を平坦に折る操作は、平面幾何学の要素を多く含むため、幾何学の学習素材として利用できる^[21]。折り手順に示された平面図を正しく読み取り、作品完成を目指すことは、形状認識力を高める訓練にもなる^[19]。また、紙という身近な素材を利用できるため、高価な教材と異なり失敗体験を許容できるという利点もある。折り紙研究の国際的な発表の場である折り紙の国際会議においても、教育事例の報告は大きな部分を占めており、論文集の約 1/5 に相当する^[74]。

芸術作品としての折り紙

鶴や兜などの伝統的な作品ではなく、近代の研究成果を取り入れた精緻な折り紙作品は、芸術分野においても認知されつつある。海外では美術館での展示会も複数開催されており、著名な作家の作品が展示されている^[32]。また、自由曲面を用いた近代建築で知られる Zaha Hadid Architects は、金属板を曲線で折り曲げたものを柱状に組み合わせたモニュメントを発表している^[3]。これは、曲線折りが生み出す形状の美しさを取り入れたものと推測される。

1.3 本研究の目的

本研究の目的は、計算機によって折り紙で作ることが可能な形状の設計を支援することである。これは、既存の CAD ソフトウェアによって達成できる課題ではない。折り紙の形状設計においては、一枚の紙から折り曲げのみで形を作るという厳しい制約が付随し、このような制約を汎用的な 3D モデリングや CAD のソフトウェアで扱うことはできないためである。

また折り紙には、一般に普及している 1 枚の紙から切り込みなしで作品を作るものだけでなく、複数枚の紙を使用するユニット折り紙や紙の切り込みを許容する場合もある。他にも、平面に敷き詰めた幾何学的な模様を折り出す平織りや、曲面を含む立体的な折り紙など様々なジャンルが存在する。そのため、それぞれの対象に特化したアルゴリズムや設計システムが必要となる。

本研究では、折り紙において最も基本となる平坦に折り畳めるもの、近年注目されている立体的な曲げを含むもの、そしてユニット折り紙と呼ばれる複数枚の紙を部品のように組み合わせるものの 3 種類の折り紙を対象として、それぞれ新しい設計アプローチおよびシステムを提案する。

提案内容は次の通りである。

- 折り畳み形状提示による平坦折り紙作品の発見手法 [81, 68]
- スリット入り平板素材の折り曲げ形状設計システム [69]
- ユニット折り紙のための組み合わせ構造設計システム [80, 70]

1 つ目は平坦に折り畳める折り紙作品のうち、既存の設計手法では困難だった折り回数の少ない作品を対象としている。計算機のパワーを活用し可能な折り畳みパターンを列挙した上で、得られたデータを作品発見に利用した。2 つ目はスリット（細長い切り込み）を含む素材の折り曲げによって立体的な形を設計するものであり、切込みを許容することで自由度の高い設計を可能にした。立体形状が満たさなければならない幾何的な制約を最適化問題として解くことで、平面に展開可能な形状を計算している。最後は、複数の小さな部品を折ってそれらを組み合わせて大きな立体作るユニット折り紙を対象としたものである。基礎となる構造が正三角形集合で表現されることに着目して、正三角形集合の立体の設計および立体がもつ性質の調査に取り組んだ。いずれも折り紙の裾野を広げる上で重要なテーマである。

なお、本稿における折り紙という用語は、1 枚の紙で形を折るものだけでなく、紙のような伸縮しない素材を使用し、切り込みと折り曲げによって形を作るもの全般を意味する。複数部品の使用も許容するが、糊付けなどの接着方法は使用しないこととする。したがって、ユニット折り紙は部品同士を重ねて折ることによる接合の仕組みを備えているため、折り紙の範疇とするが、糊付けを使用するペーパークラフトは、本稿の折り紙の定義には含まれない。

1.4 本論文の構成

本論文は本章を含めて全 6 章で構成される。以下、第 2 章では折り紙の設計に関連する研究を紹介する。第 3 章では、折り畳み形状提示による作品発見手法、第 4 章ではスリット入り平板素材の折り曲げ形状設計システム、第 5 章ではユニット折り紙のための組み合わせ構造設計システムについて述べる。第 6 章ではそれらの結論と今後の展望についてまとめる。

第2章 関連研究

古くから伝えられる伝承の折り紙作品（例えば鶴やかぶとなど）は先人の試行錯誤によって創り出されたものと考えられる。しかし試行錯誤に頼るアプローチでは、昆虫や人型のような複雑な作品を作ることが困難であった。このため、折り紙に関する数理の発展と作品の複雑化への要求を背景として、近代の折り紙では作品を設計するという概念が持ち込まれた。

本章では、まず折り紙設計において重要な役割をもつ、平坦に折りたためるかどうかの判定について説明し（2.1 節）、既存の設計手法およびソフトウェアを紹介する（2.2 節）。そして、ユニット折り紙と関係性の高い多面体について述べる（2.3 節）。

なお、本章では折り紙設計に関係の深い話題のみを取り扱う。折り紙の幾何にまつわる数学的背景や関連した問題については、「折り」に関する約 400 もの参考文献を網羅した Demaine と O'Rourke による大著^[16]を参照してほしい。

2.1 平坦折り可能性

折り紙作品として広く知られている作品のほとんどは、平坦に折りたためる作品である。例えば図 2.1 の鶴は最終的には立体的なポーズをとっているが、折り線の構造（図 2.2）は平坦に折りたためるものである。したがって、ある作品が平坦折りかどうかの判定は、折り線の構造が平坦に折りたためるかどうかで判定する。この折り線の構造、つまり一度折った紙を開いたときに現れる図は、展開図と呼ばれる。折り紙において平坦折りが重要であるのは、幾何的に特殊な性質を持つためであり、その代表的なものとして平坦折り可能性の問題がある。

2.1.1 局所平坦条件

ここでは、紙内部の折り線の交点（以後、頂点とする）に着目する。頂点が一つの展開図を考えると、この展開図が平坦に折り畳める場合、角の 1 つ置き和が 180° であり^[29]、山折り線と谷折り線の差は 2 となる^[7, 24]（図 2.3）。これらは局所的な平坦性を判定する条件であり、前者は川崎定理、後者は前川定理として知られている。与えられた展開図に対して、角度についての条件が満たされている場合、ユニークな山谷の割り当てをすべて求めることは線形時間で行えることが示されている^[25]。

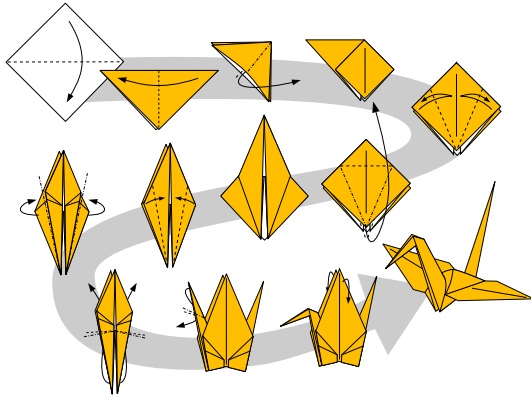


図 2.1: 鶴の折り図

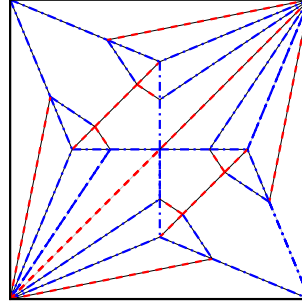


図 2.2: 鶴の展開図

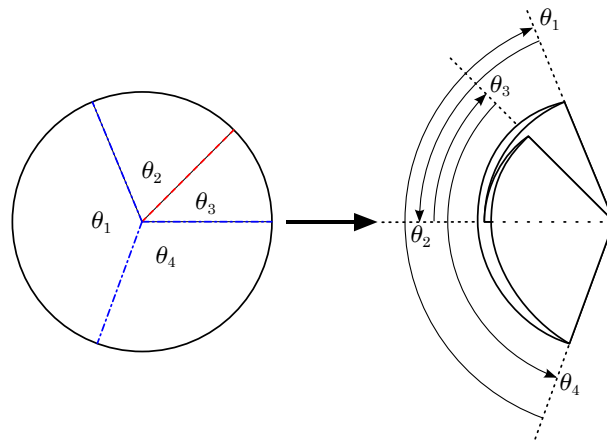


図 2.3: 平坦に折り畳める頂点: $\theta_1 + \theta_3 = \theta_2 + \theta_4 = 180^\circ$ となる

2.1.2 折り畳まれた状態の存在判定

山谷が割り当てられた展開図が与えられた時、それを平坦に折り畳むことが可能かどうかの判定はNP 困難な問題として知られている^[7]。この問題の難しさは、大域的な平坦条件、つまり可能な紙の重なり順が存在するかどうかの判定にある。局所平坦条件の確認は容易であるが、全体として平坦に折り畳めるかどうかは、解を総当たりで調べる必要がある。ORIPA^[43]は展開図から可能な紙の重なり順推定が可能なソフトウェアであり、不可能な紙の重なり順を早い段階で枝刈りすることで、高速に判定を行なっている。

2.1.3 Map Folding と Stamp Folding

上記の平坦折りについての問題は、一般的な折り紙作品のような折り線が自由な位置に配置されたものを対象としているが、より単純な問題として、格子状の折り線で折る Map Folding と、細長い帯を一定間隔で折る Stamp Folding (1 次元の折り畳み) という問題がある^[16]。後

者のような単純に山または谷の折り線が繰り返される展開図でも、紙の重なり順が異なる複数の折り畳み方が考えられる。これらは主に組み合わせ論のテーマとして取り扱われ、平坦に折れるかどうかを判定するための計算量^[4]や紙の重なり順が異なるパターンがいくつ存在するか^[72]、などが議論されている。

2.2 折り紙設計

折り紙設計とは、紙を折ることで目的とする形を作り出すことと定義する。以下では、平坦折りを対象とした設計手法と、非平坦なものを対象とした設計手法を分けて紹介する。

2.2.1 平坦折りの折り紙設計

1979年に発表された複雑でリアルな「悪魔」の作品^[40]に見られる尻尾や角、指先に至る細部まで再現された形状は、それまでの「見立て」を中心とした折り紙とは一線を画すものであった。その後、突起を作るのに必要となる円形の領域を展開図上に配置する手法に基づく circle packing (circle/river packing、tree method と呼ばれる)などの設計アプローチが考案された。これらは文献^[33]に体系的にまとめられており、このアルゴリズムを組み込んだソフトウェア (Tree Maker) も公開されている^[34]。また、作品のシルエットから設計を行うアプローチや、テッセレーションと呼ばれる決まったパターンを平面に敷き詰めるものも設計の一手法として取り上げる。

TreeMaker

TreeMaker は circle packing アルゴリズムを実装した、折り紙作品の設計ソフトウェアである。目的の形の構造を、線分のネットワーク (グラフ理論における木構造) として与えることで、その構造を折るための展開図を出力される。出力から得られるものは基本構造だけであるため、細部のデザインや作り込みは人の手にゆだねられる。折り紙を格子状に区切り、その上に四角形の領域を配置する box pleating と呼ばれる設計技法も存在するが、その基本的な考え方は circle packing と同じである。

シルエットからの設計

より直接的なアプローチとしては、折りたたまれた最終形を与えて設計することが考えられる。Shimanuki らは、動物などのイラストから骨格構造 (前述の木構造のグラフ) を抽出し、それを元に設計を行う手法を提案している^[60]。これはすでに挙げた circle packing の手法に近く、目的の構造は得られるが、その最終的な形を決めるのは折り手にゆだねられているため、直接的ではない。Demaine らは一定幅の細長い紙 (以後、ストリップと呼ぶ) を用いて、任意の領域をカバーするアルゴリズムを提案している^[15]。このアルゴリズムでは、領

域を覆うのに必要な紙の長さ、最大のストリップ幅を最適化している。さらに両面の紙を用いる場合は、特定のパターンを出すことも可能である。しかし、細長いストリップを用いていることから、正方形あるいは十分に幅のある紙、つまり一般的な折り紙には適用できない。本論文の第3章で提案する手法は、シルエットから作品を発見するものであり、正方形の紙を用いている。最終形に似ている形を得ることはできるが、厳密に一致するように折ることはできておらず、任意の形を与えて設計を行うアプローチは難しい課題といえる。

折り紙テッセレーション

特定の折り線構造を平面上に敷き詰めることで、幾何学的なパターンを折り出すことを折り紙テッセレーションあるいは平織りと呼ぶ。テッセレーションは幾何的なパターンが生み出す美しさから、折り紙の中でも注目されている一分野である。テッセレーションの展開図を設計するためのソフトウェアとしては、Tess が知られている^[5]。これは、タイル形状と中心図形の大きさ、傾き角度のパラメータから、パターンを生成することが可能である。また Lang は折りたたまれた状態が平坦に展開できる条件（つまり折りたたみの逆問題）に着目し、編み込みパターンを生成するための定式化を行なっている^[35]。

剛体折り紙

剛体折り紙とは、折り線で区切られた領域を剛体のパネルとみなし、展開図をそれらのパネルがヒンジで接続されたものとするものである。剛体折り紙は機械的なリンク構造と関連していることから工学応用の可能性が高い分野であり、大型のソーラパネルに応用された例もある。剛体折り紙の問題では折りたたみ途中の変形が重要とされる。展開された状態から平坦に折りたたまれた状態まで滑らかに変形するような構造が、折り畳みの際に必要な工程やエネルギーを少なくするという点で工学的に有用だからである。一般的な折り紙作品も多くを剛体として考えることは可能であるが、剛体折り紙の分野で取り扱われることは少ない。むしろ、テッセレーションと同じく平面を充填するパターンが議論の中心となる。

Rigid Origami Simulator^[64] は剛体折り紙の折りたたみをシミュレーションできるソフトウェアである。また、この発展である Freeform Origami^[65] は変形途中の状態を対話的に変形させることで、平坦条件を満たしたまま展開図の設計が行える。近年では、折り畳む過程も含めて、折り曲げる物自体が空間充填するような構造についても研究が行われている^[48]。空間充填構造は、正六角柱を並べたハニカム構造にも見られる重要な性質である。

折り紙建築

折り紙建築とは、ポップアップカードのように2つ折りでき、開いた状態で立体が立ち上がるものを指す。切り込みが入っている点で他の折り紙とは異なるが異なるが、これも平坦に折り畳めるため本節で触れる。折り紙建築では特に、1枚の紙に切り込みを入れて、90度を開いたときに立体が現れるもの対象とすることが多い。設計手法には、対話的な手法^[42]や

3D モデルを入力とする手法がある^[36]。一般的なポップアップカードが複数の部材を貼り合わせることを許容するのに対して、折り紙建築は切り込みを許容するものの素材は1枚のみである。この点は、本研究の第4章で取り上げるスリット入り素材を用いた形状設計とも共通しており、本研究における「折り紙」の線引きとなるルールである。

2.2.2 非平坦折りの折り紙設計

これまでに紹介したものは、すべて平坦に折り畳めるものを対象としている。それでは、平坦でない形状を扱う場合にはどのような制約や条件が必要なのだろうか。近年では非平坦な（つまり立体的な構造をとる）折り紙設計の研究がいくつか行われている。しかしながら、平坦折りの数理がほぼ整っているのに対して、立体折りには数学的に未知な部分も存在するため、対象を限定して、特定の構造の設計に特化した手法を提案しているものが多い。ここでは、展開図の構造によって、直線のみで構成されているもの、曲線を含むものに分けて説明する。

なお、以下で述べるものの他にも立体的な折り紙を対象とするものには、紙のような薄い素材の折り曲げ^[51]や破れる様子のシミュレーション^[10]、対話的に折り紙を操作できるシミュレータに関する研究^[49, 78]などがあるが、それらは形の設計を目的としたものではなく、リアルなアニメーションに利用するものである。そのため、本節では割愛する。

直線折り

立体折り紙で形を設計する場合、目的の形を得るために出てくる紙の余りをうまく処理する必要がある。これを実現する手法として、ある決まった構造を組み込んでいくアプローチが取られている。Origamizer^[66]は、三角形集合で表現された3Dモデルを入力として、それを1枚の紙で折れるような展開図を出力するソフトウェアである。各面の間に「ひだ」となる構造を組み込むことで、3Dモデルの折り紙化を実現している。また、グリッド状の迷路のような構造を折り出す手法^[14]や立方体のブロックを組み合わせた構造を折り出す^[6]などがあり、これらの研究では、ガジェットと呼ばれる複数のパターンを組み込んでいくことで、展開図を設計している。

曲線折りと可展面

曲線折りの歴史は古く、1920年代にまで遡る。現在に至るまで様々な曲線折りの立体作品が公開されてきたが、その設計の理論は未だ確立していない。曲線折りの立体折り紙として広く知られている作品に、Huffman Tower^[22]がある。空間曲線による複雑な折りを含むこの作品は、その創作手法が現在でも研究されている^[13]。以下では、曲線折りを計算機で扱うための基礎知識として、可展面（平面に展開可能な曲面）の説明をした後、いくつかの手法やソフトウェアを紹介する。

一般に、平面を曲げてできる形は可展面と呼ばれ、円錐面、柱面、接線曲面に分類されることが知られている。したがって、曲線折りの立体折り紙は可展面パッチの集合と見ることができる。このような形を設計するには、複数のパーツの集合で表現するアプローチと1枚の紙で切り貼りせずに表現するアプローチが考えられる。前者の手法を用いた研究としては、ペーパークラフトを題材としたものや^[46, 41]、実際の建築構造を考慮したオフセットが得られるような構造^[38, 55]、あるいは空間中に閉曲線をデザインすることで、その内部に可展面を生成するもの^[56]などが挙げられるが、ここでは後者の折り紙的なアプローチについて詳細を述べる。

後者のアプローチは、切り貼りを許容するアプローチに比べて形の制約が厳しく、設計はさらに難しい。Kilianらは実際に人の手で折った形を3次元計測器で測定し、その3Dモデルを可展面の集合で近似的に再構築している^[30]。しかしながら、複雑な最適化を要することからリアルタイムでの計算は行えず、対話的な設計は行えない。

現在のところ、曲面を含む立体折り紙を対話的に設計できる手法は少ない。Mitaniは断面のポリラインを入力として、軸対称な立体折り紙を設計する手法を提案している^[44]。これは直線折りで述べたような、ひだの部分を加えることで展開図を設計している。また、可展面で表現されたオブジェクトを、空間中の平面で対称に折り曲げることで形を編集するツールも公開している^[45]。これは曲線折りを対話的に設計できるが、平面で折り曲げることから、曲線が平面曲線に限定されているといえる。より単純なものとしては、紙のストリップ部分に対話的に折り曲げて形を作る手法がある^[8]。これはストリップの中心を通る曲線を入力とし、与えられた曲線を一定の間隔で区切って3次微分を計算することでストリップを生成している。この手法は、紙に切り込みを入れた形状を扱うことができるが、内部を切り抜いた穴のある形状には対応できない。Solomonらは、2次元の展開図と3次元の立体的な状態の両方を操作することで、紙の曲げ形状を対話的に編集するシステムを提案した^[63]。Zhuらの手法では、折り線の強さと影響範囲を指定することで、薄いシートがしなやかに曲がる性質を表現している^[77]。

本論文第4章でも、ストリップを対話的に編集する手法について述べる。対象とする形状を左右対称なものだけに限定して断面の形状などをパラメータとして扱うことで、大まかな立体形状を素早く得ることができる。折り曲げたストリップ形状の表現、および妥当な形状を求める最適化計算には、文献^[38]および文献^[55]で用いられているPQメッシュ(平面四辺形の集合)による表現およびその最適化手法を応用している。またメッシュを細分割していくことで、概形から詳細へと段階的に編集を行うアプローチは、Solomonらのシステム^[63]と類似している。

2.3 ユニット折り紙

ユニット折り紙とは1枚の紙で単位部品(ユニット、モジュール)を折り、それら複数個組み合わせることで一つの大きな構造を作るものである。ユニットの形の種類は単一のものから複数種を用いるものまで様々であるが、一般的なユニット折り紙では正多面体や半正多面体を組

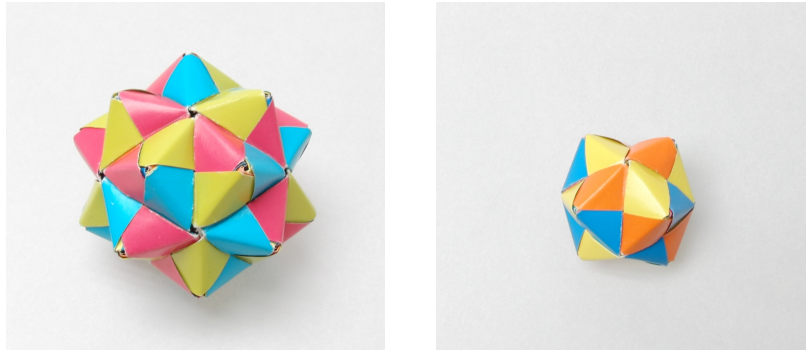


図 2.4: 園部式ユニットによるくす玉（左: 正二十面体ベース 30 枚組、右: 正八面体ベース 12 枚組）

み上げる傾向にあるため、数種類に留まることが多い。特によく知られたユニットに 図 2.4 の園部式ユニットがあり^[50]、これは正三角形（正三角錐）が基本単位となる。

個々のユニットの設計や装飾は折り紙作家の手によって様々なものが発表されているが、ここでは全体の構造（組み方）に着目する。つまり、ユニット折り紙で組み上げる構造となる多面体を設計するための手法について概説する。なお、折り紙に関連した多面体の問題としては、多面体の展開図や多面体の折りたたみなどが挙げられるが、本論文では取り扱わない。これらの問題については文献^[16]が詳しい。

2.3.1 多面体の設計

汎用的な 3D モデルを作成するソフトウェアは多く存在するが、多面体を対象としたものは多くない。Stella^[75]は多面体のための強力なモデリングツールである。正多面体などの既知のモデルが組み込まれており、それらに多面体同士の連結や双対多面体との合成など複数の操作を加えることができる。Antiprism^[2]はコマンドラインで動作する多面体の生成、編集ツールである。Antiprism は前述のような操作を複数のサブプログラムによって実行することができ、こちらも様々なモデルが生成できる。

正多面体などの多面体群は、面の形に制約がある立体と考えることができる。面の形の種類を限定することは、建築のパネルなどとして実際に利用する場合に、部品の種類が少なく済むというメリットがある。しかしながら、限られた面の形で任意の形状を表現することは難しい課題である。面の形に着目した研究としては、面の形の種類を減らすように最適化とクラスタリングを反復して行う手法^[61]や、任意の種類の四角形（ n 個の形の四角形）を用いて入力された形状を近似表現する手法^[17]がある。

本論文の第 5 章で提案するシステムは、多面体を構成する面の形を正三角形のみに制限して設計を行うものである。正三角形のみに限定した状況で可能な操作の実装と、正三角形ではない面を含む三角形集合の多面体が与えられた時に、各面を正三角形に近づけるように変形する機能を実装している。

第3章 折り畳み形状提示による折り紙作品発見の支援

折り紙設計の理論によって作品の複雑さの水準は飛躍的に向上した。伝承的な折り紙と一線を画す、精緻で複雑な折り紙作品はコンプレックス折り紙と呼ばれている。動物や昆虫などの対象物を、リアルに再現したいという欲求が、折り紙設計の技術の発展を推進してきた。その一方で、わずか数回の折り操作で作られるシンプルな作品の設計に関する研究はほとんど行われていない。例えば図 3.1 に示すような折り手に幼児を対象とした作品は、経験が豊富な折り紙作家の手によって、試行錯誤を伴う発見的なアプローチがとられているものと推察される。

本章では、折り回数の少ない作品を対象として、多数の折り紙を折りたたんだ後の形状をユーザに提示することによる、平坦折り紙作品の発見を支援する新しいアプローチを提案する。具体的には、紙を折る操作と折る回数を限定することで可能な折り畳み形状を列挙する手法と、列挙された折り畳み形状の中からユーザが入力した輪郭に近いものを抽出して提示するシステム、そして、ランダムに形状を生成して提示するシステムの3つである。これらは、発見的なアプローチによる作品創作を強力に支援する。以下では、最初に既存の設計手法の問題点と提案手法で対象とする作品を述べた後、それぞれの手法の詳細について述べる。

ところで、本章では折り回数の少ない折り紙作品を対象とするが、それらの作品に求められる特徴を知るために、簡単に作れる作品を紹介した書籍^[79]に掲載されている作品 64 点を調査した。その結果、それらの作品は平均して約 8 回の折り操作で作ることができ、最も折り回数の少ない作品は 3 回の折りだけでできていた（図 3.1 右の「ヨット」）。折り回数が多い作品でも、その手順には作品に丸みをつけるための「カドを少しだけ折る」といった操作が含まれることが多いため、作品の概形はより早い段階で決定される。以上より、数回の折り操作でも十分に作品の表現が可能であると考えられる。

3.1 既存の設計手法の問題点

circle packing およびその類似手法はシステマティックに設計が行える強力な手法である。しかし設計手法の多くは「展開図を設計する」ものであり、折り畳んだ後の形状がそのまま完成形にはならず、最終的な形を決めるのは折り手に委ねられている。つまり、折りたい作品の構造を得るのには強力であるが、折りたい形そのものを得ることは困難である。また、展開図（および折り畳んだ後の形）しか情報がなく途中の手順が存在しないことは、初級者や初めて折り紙をする人にとって理解が難しい。一般に、展開図と折り手順は一对一の関係に

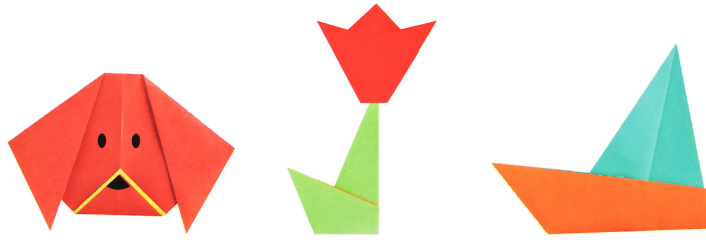


図 3.1: 幼児向けの折り紙作品の例; 左から「いぬのかお」、「チューリップ(はな と はっぱ)」、「ヨット」(文献^[79]より)

あるわけではなく、1つの展開図に対して無数の折り手順が存在する。展開図から折り手順を得るための手法も研究されている^[23]ものの、折り手順の全行程を自動的に得ることはできない。提案手法では、逐次的に紙を折り畳み、その折った後の形を提示するため、折り手順が存在し、かつ、提示された形がそのまま完成形となる。

3.2 折り畳み形状の列挙

本節では、紙を折る際に折り紙において頻繁に用いられる操作のみを使用し、さらに折り回数を4回以下に限定することで、折り畳み形状の列挙を行う手法について述べる。列挙された形状の中に含まれるであろう、我々がまだ作品として見い出していない形を発見し、これにより作品発見の支援を行う。

折り回数を4回以下としたのは、列挙される折り畳み形状の個数が、現代の計算機で十分対応可能な範囲に収まったためである。4回以下の折りで表現できる形は限定されるように思われるが、それでも可能な折り方の組み合わせ数は膨大な数となる。詳細は後述するが、4回の折り畳みより実際に得られた形状は136,284個であった。5回折った場合は、約5,400万個にまで増加したため、これを踏まえ「4回以下の折り操作で作れる形」を対象とした。

使用する折り操作を限定するのは、紙を自由な位置で折ることを許容すると、1回の折り操作にも無限のバリエーションが存在することになるためである。本手法では、参照要素(点と辺、辺は直線として扱う)を用いる次の3種類の折り方だけを対象とする。

- 点と点を通る直線で折る
- 点と点を重ねるように折る
- 直線と直線を重ねるように折る

このような参照要素の存在する折り方は、過去の研究から、全部で8通り存在することが示されている^[26, 28]。今回挙げた3通りの折り方も、これらの操作の中に含まれる。本稿では上記の操作を採用したが、これは「2つの点 p_1 、 p_2 と2直線 l_1 、 l_2 が与えられたとき、 p_1 を l_1 上に重ね、かつ p_2 を l_2 上に重ねる折り方」のような実際の折り紙では用いられない折り方を除外し、簡単な操作のみに限定するという目的がある。

また、この列挙手法においては山折りと谷折り、紙を折って開く操作のみを使用する。紙を折って開く操作とは、以降の手順で折るときに参照できる点や線をつける操作を意味し、これも「1回の折り操作」とした。折り紙には、1回の折り操作で複数の折り線を同時に折る「しずめ折り」「かぶせ折り」「中割折り」などの技法も存在するが、山折りと谷折り以外の特殊な折り技法は使用していない。なお、参照要素のある折り操作と山折り、谷折りのみを用いて作品を作ることはピュアランド折り紙と呼ばれる1ジャンルとして提唱されている^[62]。

3.2.1 折り畳み形状の生成と重複判定

ある状態における可能性のあるすべての折り線の位置は、前節で述べた3つの折り紙公理を、すべての参照要素の組み合わせに適用することで求めることができる。具体的には、2つの参照点を結ぶ線分、2つの参照点の垂直二等分線、および2つの辺の成す角の二等分線が候補となる。

続いて、このようにして求めた折り線の位置で紙を折る。紙が複数枚重なっている部分では、谷折りの場合は一番上にある面から1枚ずつ順番に折り畳み、すべてを異なる折り紙作品として扱う。これは先行研究として行った折り図作成支援ツールの開発^[67]で、折り手順の予測の際に用いたアルゴリズムと同じであり、紙が内側に折り込まれるケースが除外されるため、紙が交差するような形状が発生しない。山折りの場合は逆に、一番下にある面から順に折り畳む。「折り線をつけて戻す」操作は、すべての面に一度に折り線をつける。これら3通りの折り操作により、重なっている面の数を n とすると、1つの折り線から最大で $2n + 1$ 個の形状が生成される(図3.2)。

ここまでの処理で折り紙の形状を網羅的に生成したのち、重複するものを除外してデータベースに登録する。ここでは、ある形に対して各面の共通重心から各頂点までの距離の総和(図3.2)が等しいものを、「回転および反転して一致するもの」と判断して除外した。この方法で除外すると、輪郭は同じで面の重なり順だけが異なるものも除外されてしまうため、以降の手順において生成される形の候補が少なくなり、その結果として後述する紙の表裏の色の違いを利用した検索で適切なものが見つからなくなる可能性がある。しかし、この問題を解決するためには、紙を一旦開いて、展開図から面の重なり順が異なるパターンをすべて数え上げることが必要となる。異なる手順で同じ展開図が得られる可能性もあるため、それらの重複判定を含めると、データベースに格納される形の一意性を維持するのは、難しい問題と言える。今回は折った後に現れる輪郭の形に主眼を置いたためこの方法を採用したが、今後検討すべき点と言える。

なお、新しい形を格納する際には1手前の形が格納されている場所の参照を保持するようにした。これを順に辿ることで、後から折り手順を知ることが可能である。

3.2.2 折り畳み形状の列挙結果

折り紙形状の列挙によって得られた結果を表3.1に示す。今回は4回までの折り操作としたため、4回目の折り操作では「折り線をつけて戻す操作」によるデータを含めなかった。表3.1

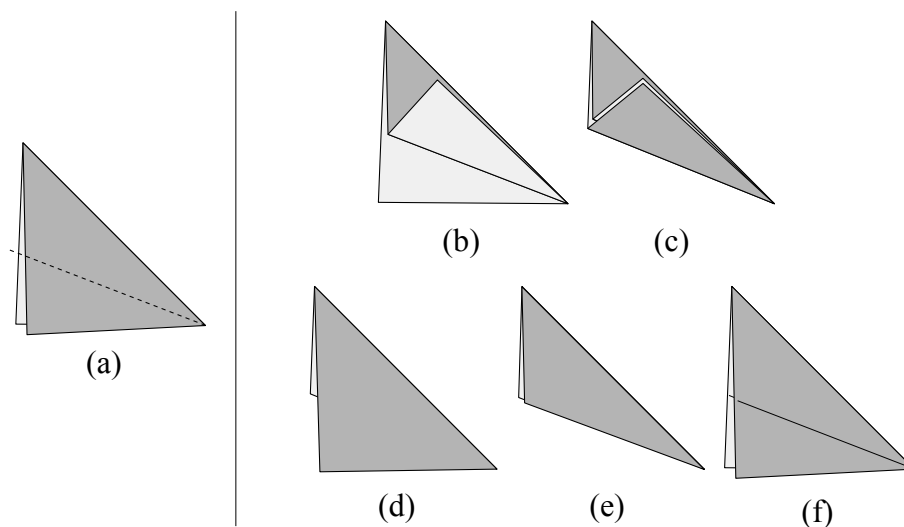


図 3.2: 1 つの折り線から得られる形状; (a) では 2 枚の紙が重なっており、谷折りで (b) と (c)、山折りで (d) と (e)、折ってひらく操作で (f) が得られる

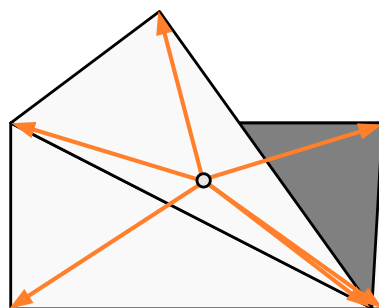


図 3.3: 同一形状の判定に用いる値の算出

からは、折り回数が増えるに従ってデータ数が指数関数的に増加していることがわかる。実際に生成された形状の一部は付録に記載した。これらの形状の中からは図 3.4 のような作例が発見できた。

表 3.1: 折り回数とデータ数および構築時間

折り回数	データ数	構築時間
1	4	47 (ms)
2	37	423 (ms)
3	1,507	1421 (ms)
4	136,284	122.747 (s)

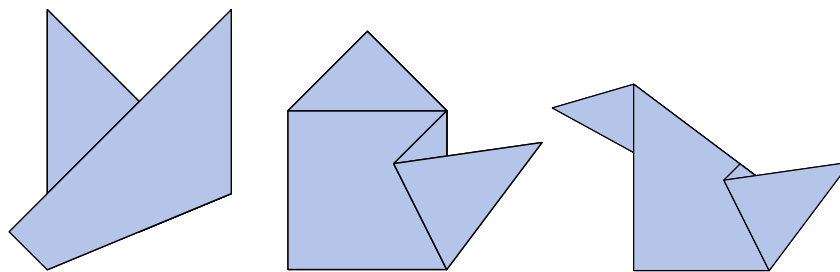


図 3.4: 列挙した形状から発見した作品の例: 左から、3 回折りのキツネ、4 回折りのティーポット、4 回折りの鳥

3.3 ユーザ入力に似た作品の検索システム

形状列挙によって膨大な数の折り畳み形状が得られるが、この中から目的とする形を見つけ出すことは時間のかかる作業である。そこで、ユーザが入力した形状に似たものを抽出することで、作品発見を支援するシステムを構築した。提案システムの概要を図 3.5 に示す。上側は計算機があらかじめ行う処理を、下側はユーザが入力を与えた後の処理を示している。本稿で提案するシステムの流れは次のとおりである。

- 4 回以下の折り操作で得られる折り紙形状をすべて列挙しデータベースに格納する。
- ユーザは折りたい形状を多角形として入力する。また、紙の表裏の色の違いを考慮した、2 色の多角形の集合を入力とすることも可能である。類似形状の判定には用いられないが、オプションで目および鼻などのパーツも配置可能である。
- 入力された多角形に形が類似した折り紙作品を検索し、類似度順に上位複数個の検索結果をユーザに提示する。入力に目や鼻のパーツが配置されている場合は、同じパーツを配置した結果を示す。
- ユーザは提示された折り紙作品を選択することで折り手順を確認できる。

形状列挙は前述の手法によるものであり、以降では提案システムのインタフェースと類似の評価方法について述べる。

3.3.1 検索システムのインタフェース

図 3.6 にシステムの実行画面を示す。図 3.6 左上は入力画面であり、ユーザは輪郭を表す多角形をマウスクリックにより入力する。これは各頂点を順番にクリックしていくことで行う。また、オプションで色の付く多角形領域を追加できる。これにより、折り紙の色のついた面が現れる部分を指定できる。さらに追加で、顔のパーツを配置できる。ツールバーにある目や鼻のボタンをクリックし、選んだパーツの位置をカーソルで指定し、マウスホイールによ

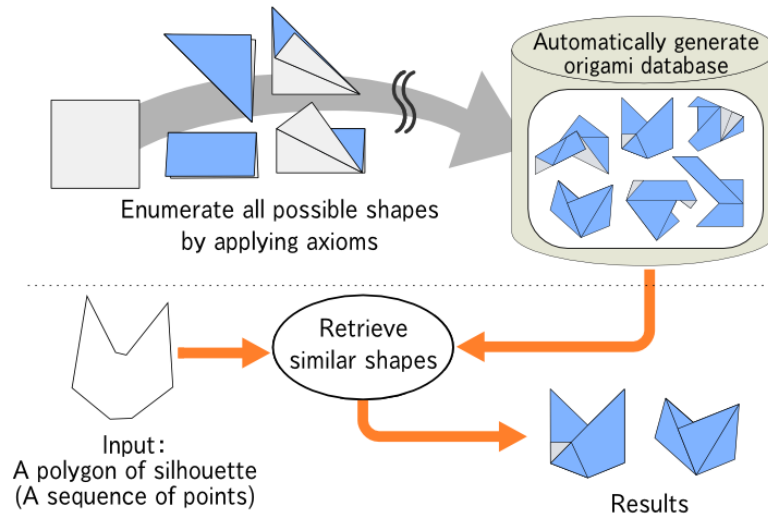


図 3.5: システム概要

て大きさを調整できる。ここで配置したパーツは検索結果に重ねて描画される。顔のパーツはシステムの起動時に画像ファイルから読み込まれるようになっているため、他の画像を用意すれば新しいパーツとして追加することが可能である。

顔のパーツを入力できるようにしたのは、折り紙作家の実際の作例に見られる「目や鼻の追加」の仕組みも考慮したためである。書籍^[79]の作品を見ると、幼児が楽しめるよう、作品のモチーフは動物であることが多く、最後に顔を描くものが多かった。細部まで作りこむのではなく、簡単に作れることを重視した結果と思われる。特に、完成後に目を描く作品が多く、調査したものの半数以上がこれに該当した。このことから、目や鼻の配置がより実用的なシステムにつながると考えた。

検索結果は図 3.6 中央のように一覧表示される。表示された中から 1 つを選択することで、その形状を折るための折り手順が検索結果は図 3.6 右のように表示される。

3.3.2 形状の類似度評価

形状の類似度判定には HuMoment^[37] と呼ばれる形の特徴を示す指標を用いる。この指標を用いた理由は、頂点と数式で表されたベクタ画像に対して適用可能であり、折り紙形状データに適しているということと、画像の拡大縮小・平行移動・回転に対して不変という性質を持つためである。この指標は画像処理で用いられる OpenCV の cvMatchShapes 関数にも実装されており、本システムもこれと同様の実装を行った。具体的には、入力した輪郭形状と折り紙形状からそれぞれ HuMoment を算出し、それらの値の差分を類似形状の判定に用いてい

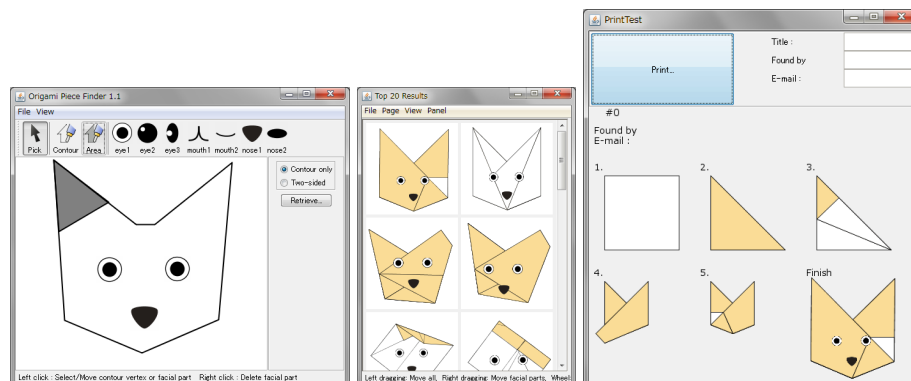


図 3.6: システム実行画面（左: 入力ウィンドウ、中央: 結果表示ウィンドウ、右: 折り手順表示ウィンドウ）

る。入力された多角形とデータベース内の全折り紙データを比較したのち、最終的に上位 20 個を出力するものとした。

3.3.3 検索システムの実行結果

提案システムで構築したデータベースの詳細と、入力としていくつかの輪郭多角形を与えたときの出力結果を示す。さらに、目や鼻などの顔のパーツの表示について考察する。なお、システムの実装には Java と SQLite を用いて、Core i5 2.67GHz, 4GB RAM を搭載した PC 上で動作させた。

例題として 3 つの多角形を入力として検索を行った。その結果および実際に折った作品を図 3.7 に示す。結果に示した折り紙形状は、各入力に対して得られた上位 20 個のうち、筆者らがより似ていると判断した 3 つである。図 3.7 の結果からは入力に似た折り紙作品が得られていると言えよう。入力形状と合わないような折り紙形状が出力される場合もあるが、最終的な判断はユーザが実際に見て行うため、ある程度類似した形状が抽出できれば十分だと思われる。上位 20 個程度まで絞り込めれば、人の目で容易に判断することができた。ただし、4 回という折り操作ではカドが多い形状を折り出すことが難しく、例えば「星型」などの入力に対しては似た形状は発見できなかった。

検索に要した時間は約 20 秒であった。検索時間の内訳は、輪郭線の場合、折り紙データとの比較に 18 秒、その上位 20 個に対して向きを合わせるための計算に 2 秒かかっている。なお、各折り紙形状の HuMoment はデータベースを生成する際にあらかじめ計算しておき、検索の際には計算済みの値を用いた。検索を高速化する手法としては検索の並列化や、似た形状をあらかじめグループ化しておきデータベースを階層的な構造にすることが考えられる。

紙の表裏を考慮した検索

図 3.8 には、紙の表が現れる部分を指定した入力に対して、輪郭だけで検索した場合と表裏を考慮して検索した場合で、最も似ていると判断した出力結果を示した。結果としては、共通して出力される形状はわずかで、異なる形状が多く出力された。図 3.8 中央は表裏を考慮した検索結果の上位 20 個には含まれず、また、逆に図 3.8 右は輪郭のみの出力結果に含まれなかった。図 3.7 に示した例でも、一部の色を指定することで違った結果が得られると思われる。表裏を考慮した検索の精度をさらに高めるためには、次の 3 つの方法が考えられる。

- 折り紙を裏返した場合も比較する
- 重複の削減手法を改良する
- 他の折り方を実装して候補を増やす

折り紙形状の中には、表から見た場合と裏から見た場合で、色の出ている部分が違う形状がある。そのため、反転した形状も比較すればそれだけで違う結果が得られる可能性がある。特に、折り紙においてよく用いられる「中割り折り」を採用することは、表裏を考慮した検索に非常に効果的であると考えられる（図 3.9）。

顔のパーツの表示について

図 3.10 には目のある場合とない場合の画像を示した。目があることで、動物としての認識のし易さが大きく向上する。本研究で取り上げたような簡単な作品では、回転させるだけで違う作品に見えることや、何に見えるかが人によって異なる場合が多い。顔のパーツの表示はそういった曖昧さを解消するための重要な要素であると考えられる。今回は顔のパーツのみを指定する機能を実装したが、顔だけでなく、例えば「くるま」や「いえ」などの作品であれば窓を描く場合もある。そのため、顔のパーツの位置を指定するだけでなく汎用性の高いペイント機能があれば、検索結果での見やすさをより向上させることが可能である。

3.4 ランダムな折り畳み形状提示

本節では、ランダムな折りを含む形状提示による作品発見システムについて述べる。形状列举に基づく手法は、指定された折り方のみで得られる形状をすべて調べ上げることができる一方で、折り回数が多くなると列举が困難になるという問題があった。また、参照要素を使用しない曖昧な折り（ぐらい折り）が使用できないという点では、形状のバリエーションに乏しかった。そこで、ぐらい折りを含む折り畳み形状をランダムに提示することで、これらの問題を解決する。

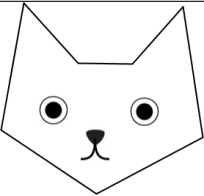
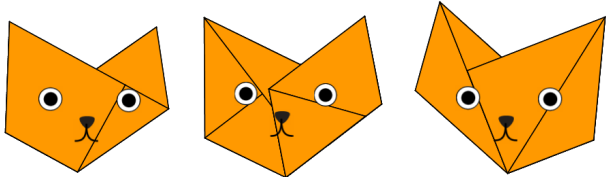

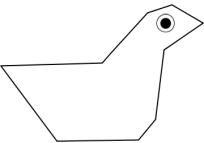


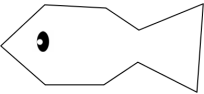
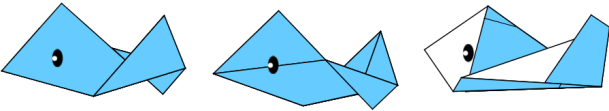

入力	出力結果	実際に折った作品
 猫の顔		
 鳥		
 魚		

図 3.7: 入力形状と検索結果、および実際に折った作品

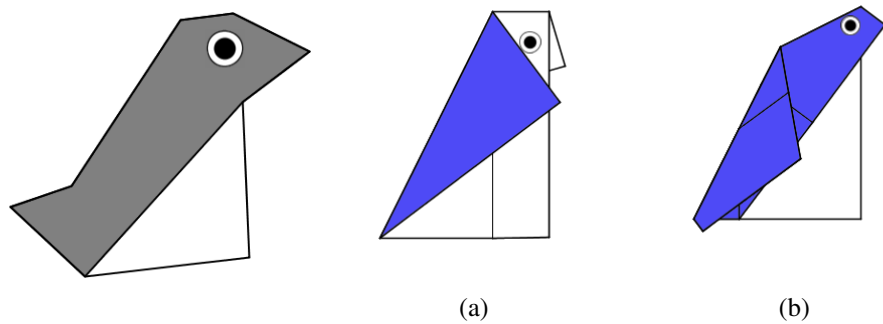


図 3.8: 紙の表裏を考慮した検索例: (a) 入力; (b) 輪郭のみを用いた検索結果; (c) 表裏を考慮した検索結果

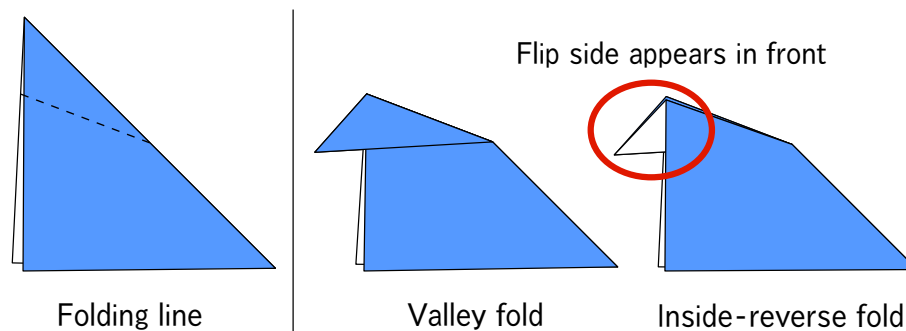


図 3.9: 中割り折りによる色の変更効果

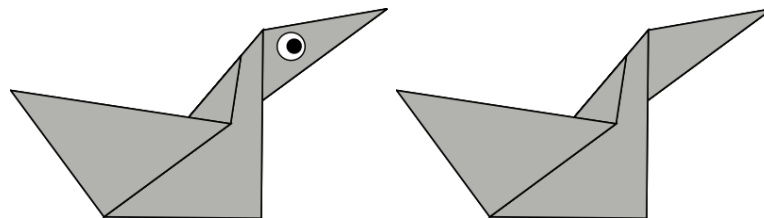


図 3.10: 目の有無による認識の違い

3.4.1 システムの概要

提案システムはランダムに紙を折った形状を自動生成して、図 3.11 のように複数個同時にユーザに提示する。ユーザはそこから欲しい形を選択することで作品の発見を行う。提示された中に欲しい形がない場合は、形状生成を再実行する。この形状生成プロセスはランダム性を含むため、毎回異なった形状が提示される。

本システムの折り畳み形状生成はぐらゐり折りを含むことができるため、前述の列挙手法よりも折り操作が拡張されている。ただし、完全にランダムに紙を折るだけでは、折り紙作品らしい形状を自動生成することはできない。したがって、列挙手法でも用いた、カドとカドを合わせるといった折り操作も同時に使用する。ランダムに折る際は、どのように折る位置を決めるかが重要となるが、本稿では2通りの手法を実装した。1つ目は参照要素を使用して得られた位置をランダムな方向にずらす手法、2つ目は紙のカドをフチに合わせるといったような、自由度のある折り操作を採用することで、ランダムではあるが整った形状を生成する手法である。図 3.12 は自由度のない折りと自由度のある折りの具体例である。図 3.12 右のような自由度のある折りを3通り追加し、合計で6通りの折り操作を使用する。また、折り畳み形状自動生成の拡張例として、左右対称な折り畳み形状の生成も提案する。こちらは山折りと谷折りだけでなく、開いてつばすような折りも使用する。

以降では、2通りの折る位置の決定方法および左右対称な形状生成、インタフェースについて述べる。提案システムでは、発見した作品をデータベースサーバへ登録することで作品の共有が可能であり、これを通して発見した形状の特徴についても考察する。

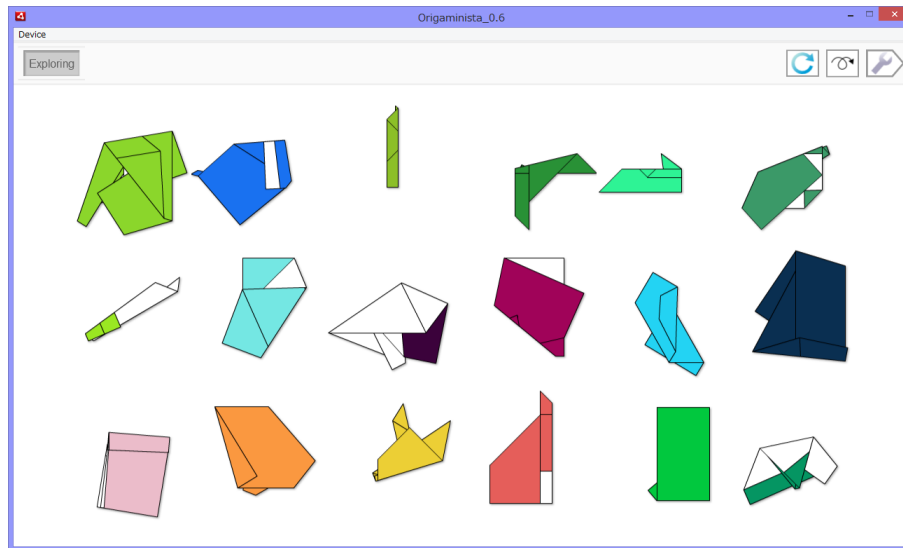


図 3.11: 提案システムのインタフェース

3.4.2 ランダムな形状生成

折り畳み形状をランダムに生成するには、前述の列挙手法のそれぞれの手順においてパラメータをランダムに決定すればよい。具体的には、折りの位置と折り畳み方向（山折りまたは谷折り）、そして折る枚数である。これらのパラメータを決定することで紙を1回だけ折った形状が得られ、さらに指定回数繰り返すことで最終的な折り畳み形状を得る。ただし、列挙手法での生成アルゴリズムとは異なり、自由度のある折り、つまり参照要素のみでは位置が決まらない折りも採用する。

3.4.3 折りの位置

列挙手法では折り紙公理に基づき折りの位置を決定した。本手法でも同じ公理を使用するが、これは最初に使用する公理をランダムに決定し、さらに必要な参照要素もランダムに選択することで実現する。その後、折る位置をランダムな方向にずらす、または、自由度のある折りでは残りのパラメータをランダムに決定することで、最終的な折る位置を得る。

ランダムな方向へのずらしは、参照要素の位置から計算された線分の端点をそれぞれ異なる方向に一定量だけ移動することで実現した。したがって、2つの角度と移動量の合計3つのパラメータが必要となる。移動方向によっては、結果として紙が折れなくなってしまう場合も考えられるが、その場合は公理の決定からやり直すこととした。結果は後述するが、この方法によって少ない折り回数でも様々な作品を発見することができた。一方で、参照要素のみで決定されるものを自由度0の折り操作としたとき、このランダムなずらしは自由度3でありランダム性が高いこと、また、パラメータによっては元の位置とほぼ無関係になってしまい、参照要素を使用する必要性が薄いという問題があった。

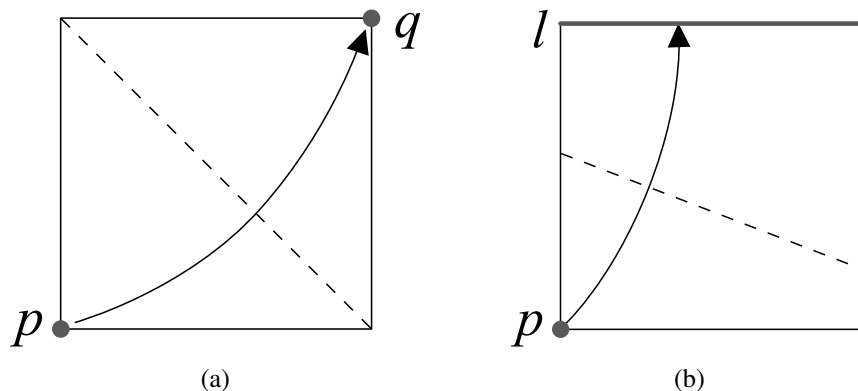


図 3.12: 自由度のない折り操作と自由度のある折り操作: (a) カドをカドに合わせるように折る; (b) カドを紙のフチに合わせるように折る

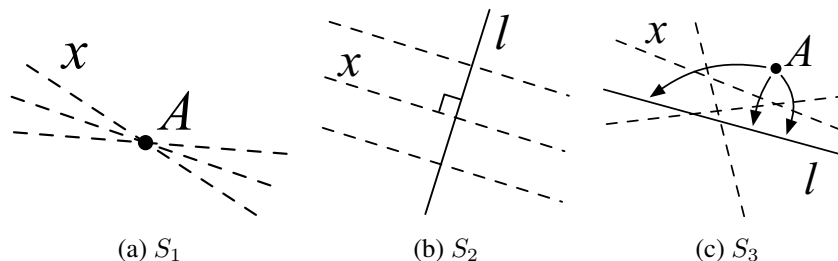


図 3.13: 1 自由度の折り操作: (a) 点を通るように折る; (b) 直線に垂直に折る; (c) 点を直線に乗せるように折る

そこで、ランダムなずらしの代わりに、自由度のある 3 通りの折り操作を追加した。それぞれの操作は図 3.13 に図示する。

(S_1) 点 A を通るような直線 x を折る

(S_2) 直線 l に垂直な直線 x を折る

(S_3) 点 A を直線 l 上に乗せるような直線 x を折る

これらの操作は Alperin と Lang によって定義された、5 つの単一の折りを与える点と線の組み合わせ (*alignments*) のうちの 3 つである^[1]。平面上の任意の直線は 3 つの自由度を持つが、これら 3 つの操作は 1 自由度しか持たない。したがって完全にランダムに折るよりも整った折り畳み形状を得ることができる。それぞれの操作で残っているパラメータは、ランダムに決定する。各操作における残りのパラメータは、 S_1 では角度、 S_2 と S_3 では直線（実際には紙のフチや既に存在している折り線）上の位置である。なお、いくつかの折り紙公理は 2 つ以上の解を持つが、これらの解は離散的なものであるため、本稿では自由度のない折り操作として扱っている。

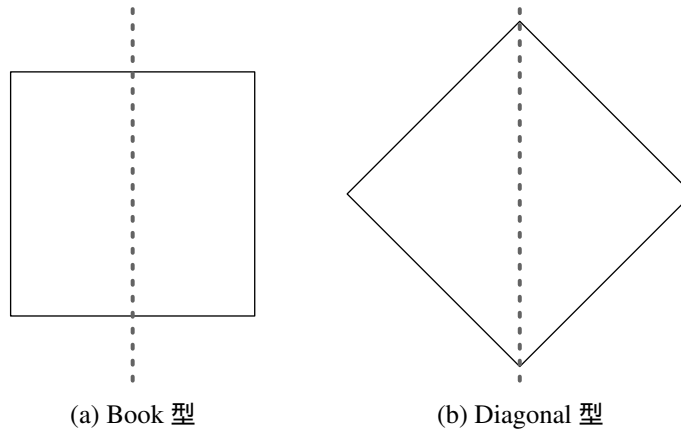


図 3.14: 2 種類の対称軸

3.4.4 折る方向と折り畳み可能な枚数

折る方向は山折りまたは谷折りのどちらかを選択する。折る枚数については列挙手法とほぼ同じアルゴリズムを用いるが、すべての候補の列挙はしない。最初に、紙を折る枚数 k を 1 から m までの間で選ぶ。 m は紙の現在の状態における紙の重なった枚数である。そして、谷折りの場合、紙の上から k 枚を折ることを試みる。この時、折る位置とすでに存在している折り線が交差している場合など、指定した k 枚のみで折り畳めない場合がある。この場合は、次の層も合わせて折ることを試みて、正しく折り畳めるまで繰り返すことで折り畳み形状を得る。指定した枚数で紙が折れるかどうかは、内部頂点の平坦折り条件を調べることで判定することができる。このように最上位の層から順に折り畳んでいく限り、最終的な折り畳み形状は常に大域的にも平坦折り可能となる。谷折りの場合は、この手順を一番下の面から行う。

3.4.5 左右対称な折り畳み形状生成

本節では、ランダムな折り畳み形状生成を拡張した、左右対称な折り畳み形状生成について述べる。この生成アルゴリズムでは、対称軸の種類を新たなパラメータとして追加する。対称軸のとり方は 2 種類あり、正方形の紙に対して長方形に折るものと対角線で折るものに分けられ、それぞれ book 型、diagonal 型と呼ばれる (図 3.14)。このいずれかを、紙を折り始める前にランダムに決定する。

紙を左右対称に折るためには、単一の折り線のみを折る単純な操作だけでなく、複数の折り線を同時に折るような複合的な操作を使用する必要がある。このアルゴリズムでは「紙を左右対称に折る 1 回のステップ」において以下のいずれかを使用する。

- 複数の折りのセット
- 開いて潰すような折り

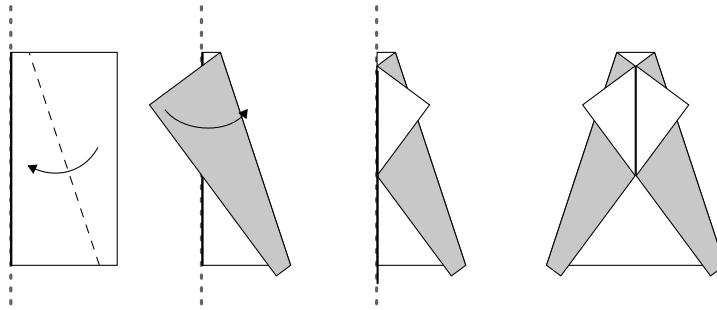


図 3.15: 折りが対象軸と交差しない場合、全体の折り操作は最大 2 回の折りになる

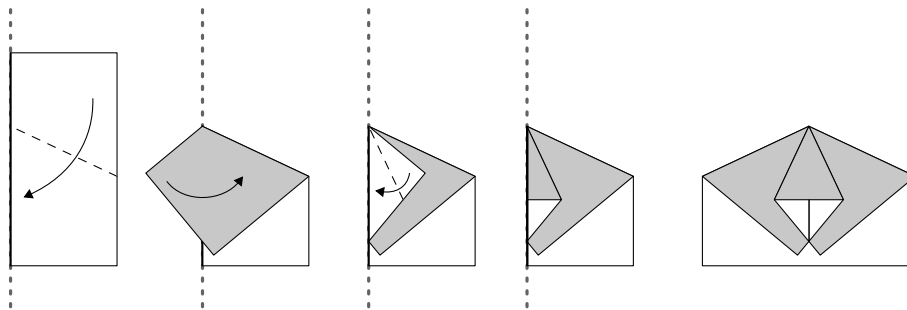


図 3.16: 折りが対象軸と交差しない場合、全体の折り操作は開いてつぶす折りになる

を使用する。ただし、どちらの折り操作においても左右対称に折るために必要なすべての折りは、一つの折りを元にして計算される。したがって、折る位置の決定方法、折り畳み方向、折る枚数の決定は、すべて前節と同じ計算を用いることができる。以降では、上記 2 通りの折り操作の詳細を説明する。

折る位置を決定した後は、得られた折りと対称軸との位置関係によってどちらの折り操作を使用するかを判定する。折りと対称軸が交差していない場合は、複数の折りのセットを使用する。図 3.15 は交差していない場合の折り操作を右半分のみ示している。ここで重要なのは、最初の折りによって移動する部分と対称軸との位置関係である。移動する部分が対称軸と交差していないならば、反対側も同様に折るだけで、左右対称な形状となるが、交差しているときは、その部分を対象軸上でさらに折り返す必要がある。このため、全体としては、最大で片側 2 回の折り操作となる。

交差している場合は図 3.16 のように開いてつぶすような操作を使用する。最初の折りによって移動する部分は、折りが対象軸と垂直でない限り、対称軸と交差する。この部分は軸によって折り返され、さらに、軸と辺が成す角の 2 等分線によって、辺が軸と重なるように折り返される。その結果、全体の折り操作は開いてつぶす折りとなる。折りが対称軸に垂直なときは、単一の水平な折りとなる。

これら 2 通りの折り操作を使用する限り、折り畳み後の形状は必ず左右対称になる。ただし、左右対称に折るための操作は他にも存在し、これら 2 つのみではない。

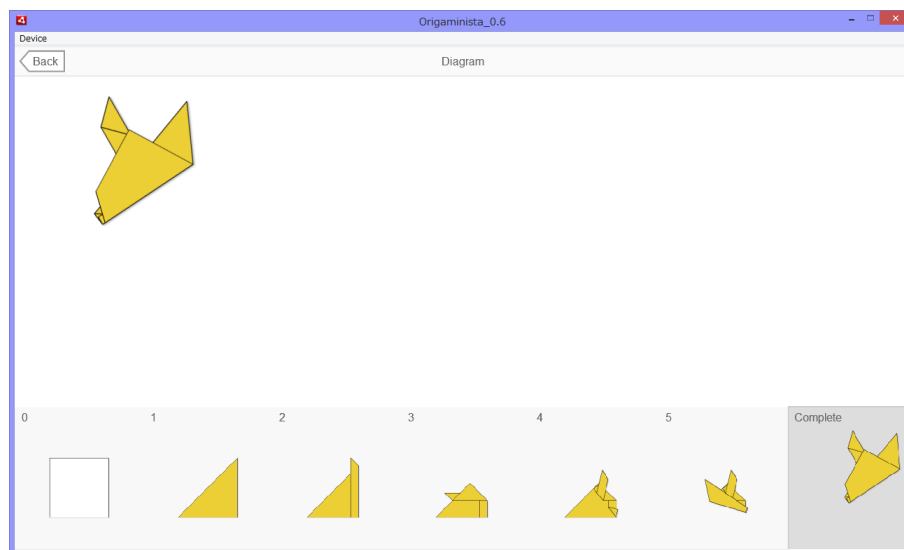


図 3.17: 折り図の表示画面

3.4.6 インタフェース

システムのインタフェースは図 3.11 に示す通りであるが、ここではユーザ操作について紹介する。ユーザは提示された中から動物などの形に見えるものを探し、気に入ったものがあれば、図 3.17 のように折り図を表示できる。見立ての成立する形状がなければ、形状生成を再度実行することができる。それ以外には、次の操作が可能である。

- 折り回数の指定
- 折り畳み形状の生成
- 回転、反転、紙の色の変更
- 折り図を表示し、タイトルをつける
- オンラインデータベースへの作品登録
- 形状生成における折り始めの状態の指定

最後の折り始め状態の指定は、形状生成プロセスの初期状態を変更するものである。通常は正方形の何も折られていない状態から折り始めるが、表示した折り図のいずれかの状態を選択することで、その選択した状態からランダムに折った形状を生成できる。言い換えると、最初の数ステップを固定する機能である。

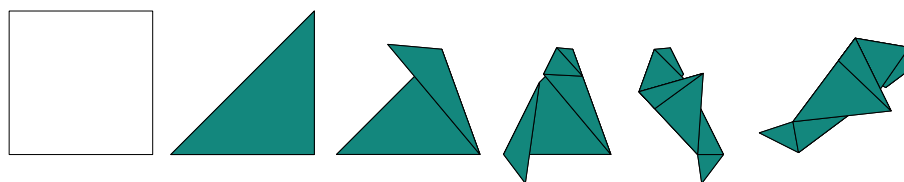


図 3.18: 自動生成したマナティの折り図

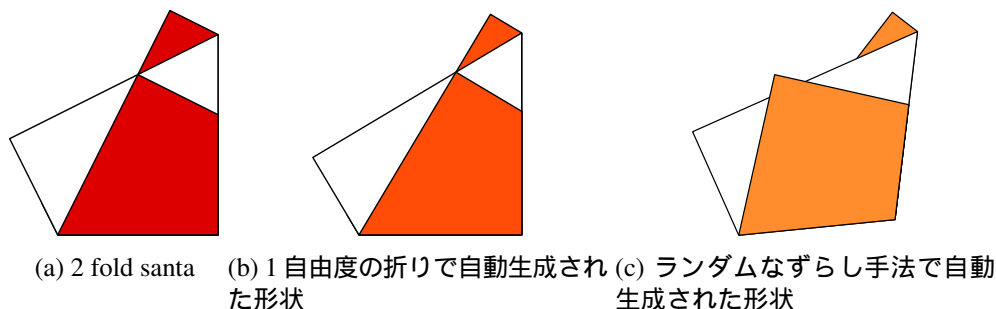


図 3.19: Paula Versnick 作「2 fold santa^[73]」とシステムで自動生成された形状

3.4.7 システムの実行結果と発見した作品例

提案システムは Action Script を用いて Flash アプリケーションとして実装した。オンラインの折り紙作品登録用データベースは MySQL と PHP で実装した。形状生成に要する時間は折り回数によって変化するが、6 回折りの形状を 20 個生成するという条件で、Intel Core i7 2.6 GHz CPU の PC 上でリアルタイムに動作した。紙を 1 回折る毎にレイヤー数は約 2 倍に増加するため、折り畳み可能なレイヤー数を計算する時間は指数関数的に増加する。図 3.18 は我々が発見した作品例である。作品発見を試行した結果、経験的には、10 回から 20 回程度の再生成ごとに 1 つの作品を見つけることができた。

また、折り紙作家の手によって設計された既存の作品と類似した形状の発見を試みた。図 3.19 は折り回数の少ない作品として有名な Paula Versnick^[73] 作の“2 fold Santa”と、それに類似した形状を示している。“2 fold Santa”の最初の折りは、カドを紙のフチの中央付近に置くような、1 自由度の折りである。0 自由度の折り操作のみでこのような折りを得るには、紙を一度折って開いて戻すという折り目をつける操作が必要になるため、折り手順が多くなる。1 自由度の折り操作を使用することで、少ない折り回数で形のバリエーションを増やすことができる。ランダムなずらし手法では、このような整った形状が得られることはほとんどない。

ウェブ上のデータベースに登録された作例

図 3.23 はオンラインデータベースに登録された作品である。これらの作品は、ランダムなずらし手法を使用した生成プロセスで発見されたものである。これらの作例を観察した結果、共通する特徴を持つ 3 つのグループを発見した。図 3.23、1 行目の 3 作品はすべて「鳥」の作

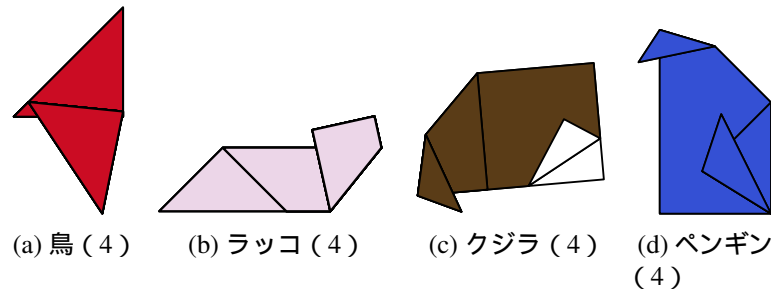


図 3.20: 子供が発見した作品例

品である。これらは、一番最初に三角形に折るとそのカドがくちばしに見えることから、発見しやすいと考えられる。2つ目のグループは「フードを被った人」という作品で、図 3.23 の2行目にある4作品がこれに含まれる。この作品群はインサイドアウト表現によって、紙の表でフードの部分を、紙の裏で人の顔を表している。インサイドアウト表現はシンプルな折り紙作品と複雑な作品のどちらでも使用される技法であるが、特にシンプルな作品においては、少ない折り回数で豊かな表現を実現できる利点がある。3つ目はボートとヨットである。これらはシンプルな折り紙作品における典型的な創作対象であり、子供向けの折り紙の本に頻繁に見ることができる。ヨットの作品では、インサイドアウト表現が使用されている。上記以外の作品に対しては、特徴を発見することができなかった。言い換えると、これ以外の作品は発見しづらく希少な作品であるといえる。

子供向けのワークショップでの作例

5歳から7歳までの子供4名を対象にワークショップを開催した。図 3.21 は会場の様子である。システムの操作説明を10分間行った後、40分の実習で平均7つの作品を発見することができた。図 3.20 は発見された作品のうち4つを示している。提案システムが子供にとっても十分に使用できることが確認できた。しかしながら、自動生成された折り図には矢印などの折り方を示す指示記号や紙の重なり関係を示すズレ表現が含まれていないため、図が把握しにくいという問題もあった。

左右対称な形状生成の作例

図 3.22 は左右対称に折るアルゴリズムによって得られた作例である。これらの作品は「紙を左右対称に折る操作」を2回だけ繰り返したものである。前述のとおり、紙を左右対称に折る操作は複数の折りを含むため、合計で何回の折りが必要かはそれぞれ異なる。左右対称に折るだけで、人がデザインしたかのような整った形状を得ることができた。形状生成アルゴリズムに追加の制約を与えることは興味深い課題であり、左右の対称性だけでなく、回転対称などを考慮することで、また異なる魅力的な形状生成が可能になるとと思われる。



図 3.21: ワークショップの様子

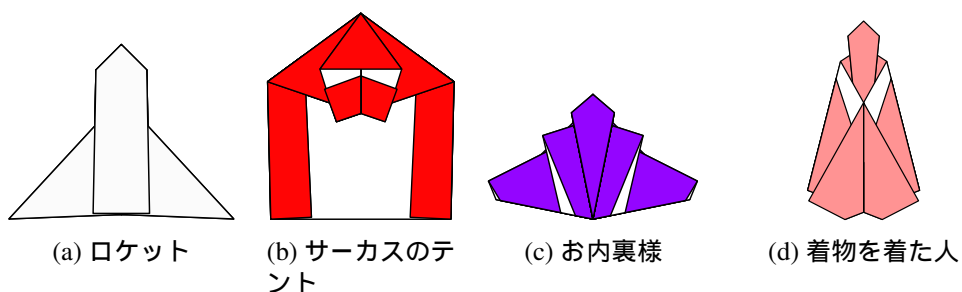


図 3.22: 左右対称な形状生成によって発見された作品例

3.5 考察

本章では、多数の紙を折り畳んだ形状をユーザに提示し、その形や色を他の対象に見立てることで作品の発見を行う手法について述べた。決して理論的ではないが、計算機のパワーを活用した全く新しいアプローチである。システムのウェブ公開やワークショップを通して、提案手法が少ない折り回数の作品発見に対して有効であることが確認できた。

この形状提示による作品発見の大きな課題としては、人がどのように折り紙の形を認識するか、という点がある。オンラインで収集した作品群については共通する特徴についてのみ簡単に触れたが、発見しやすい作品と、折り回数やそれに使用した折り操作との関係については未調査である。今後、より多数の作品データを解析することで、少数の折り回数で完成する作品についての新たな知見が得られることが期待される。また膨大な作品のデータベースが得られれば、計算機による折り紙作品の自動認識も興味深いテーマになると考えられる。

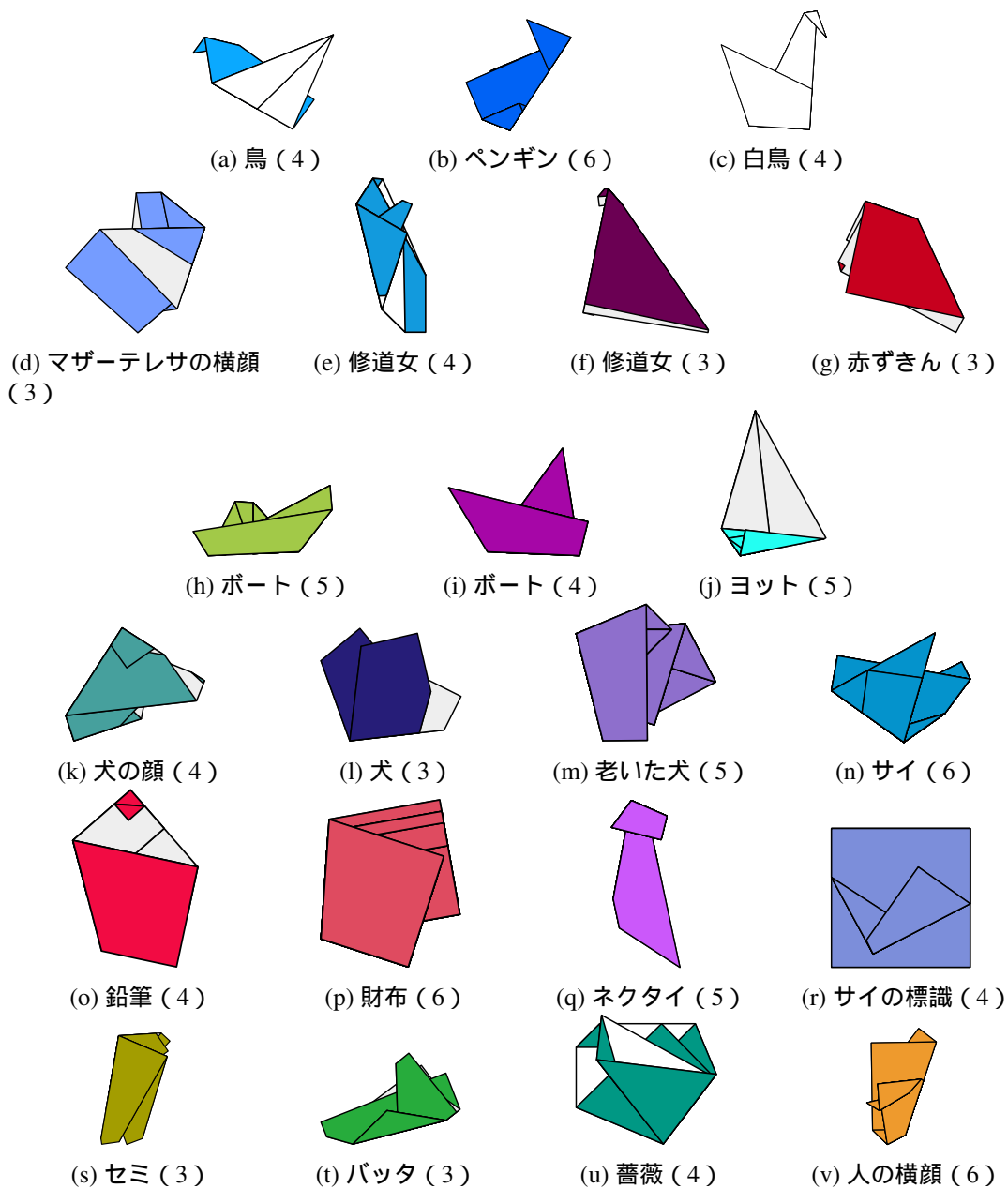


図 3.23: オンラインデータベースに登録された作品例 (括弧内の数字は折り回数)

第4章 スリットを含む平板素材の曲げ形状設計

第2章の関連研究で紹介したように、これまでに複雑な折り目を入れることで立体的な形状を作り出す各種の手法が提案されてきた。しかしながら、そもそも折りたたみのみでできる形は制約が非常に厳しく、自由に形状を設計することは難しい。

では、形の自由度を上げるにはどのようなアプローチがあるだろうか。一つは、複数の素材を組み合わせることが考えられる。これはペーパークラフトのようなアプローチであり、個々の部品は平坦な素材から切り出すことが可能という条件の元で、複数の部品を接合することで得られる最終的な形を設計する。もう一つは、切り込みを入れることである。平板素材に細い切り込み（以降ではスリットと呼ぶ）を入れることで、1枚の素材からでもより複雑な形状を作り出すことが可能になる。ペーパークラフトほどの形の自由度はないが、スリットを入れない場合に比べれば自由度は大きく向上する。スリットを用いる最大のメリットは1枚の素材から形状を設計できることであり、工学的には部材の接合のための工程が不要であるという利点がある。

そこで、より条件の厳しい後者の切り込みを入れることに着目し、スリットを入れた平板素材の折り曲げによる設計を試みる。その例題として図4.1に示す椅子を取り上げ、このような形を計算機上で構築することを目的とした。今回対象とするものは、長方形の部材にスリットを入れただけのものであり、素材を有効に活用することができる。多くの既存CADシステムでは立体形状の可展性や展開後の形状を取り扱うことが難しく、この椅子もデザイナーが紙を用いて試行錯誤の繰り返しにより設計したものである。

提案手法は、図4.1右に示すような長方形に複数の平行なスリットを入れた素材を対象とする。なお、以降では図4.1右の細長い帯状の素材領域をストリップと呼ぶ。この形状設計の問題は、各ストリップの形を如何にして決定するかという問題に帰着できる。この問題の本質的な難しさは、ストリップの長さが固定であると同時に、3次元空間における両端の位置にも制約が付くことである。一端が自由端であれば、各ストリップは自由に曲げることができるが、両端の位置が固定されているために幾何的な制約問題をストリップ毎に解く必要が生じる。

以下では、上記の制約条件を満たすための最適化手法および設計システムについて述べる。提案システムで設計した形状を、板状の素材と3Dプリンタを用いて実際に制作することで、正しく設計が行えることを確認した。提案するシステムを用いることにより、新しいデザインの創発や実際にものづくりを行う際の支援が期待される。

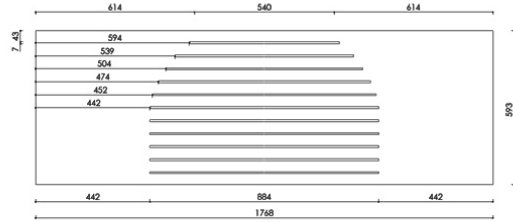


図 4.1: スリットを入れた素材の折り曲げによる椅子とその展開図 (成瀬・猪熊建築設計事務所 [52])

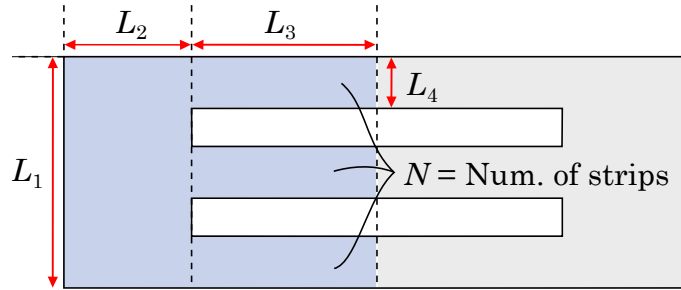


図 4.2: 椅子の展開図; L_1 から L_4 までの連続値およびストリップの数 N によって定義される

4.1 対象とする素材の形状

本稿で提案するシステムでは、図 4.2 のように長方形のスリットが平行に入っており、かつ左右対称な形状を対象とする。そのため、以降では片側のみ取り扱う。提案システムでは Liu らの手法 [38] のように、ストリップの形状を複数の平坦な四角形領域の集合 (PQ-Strip) で近似表現する。ストリップの曲げは、稜線の位置および稜線の折れ角によって決定される。以降では、この稜線を折り目とよぶこととする。 n 分割されたストリップは $(n+1)$ 本の折り目を持つ。あるストリップ上の折り目 E_i は、ストリップの縁に位置する 2 点 $(x_{L,i}, x_{R,i})$ の組と折り角度 θ_i で表される (図 4.3)。本システムでは、意図した形となるようにこれらの値を決定することとなるが、その際、3 次元空間におけるストリップ端の位置が制約条件となる。また、本システムでは得られる形状がユーザの指定した形に近いこと、およびストリップが滑らかな曲面を成すことも条件として加える。提案システムでは、それぞれをペナルティ関数で表し、ペナルティを最小化する最適化問題として扱うことで、最終形状を決定する。両端の位置制約についても、位置から離れた時に大きなペナルティを与えることで、統一的な枠組みで他の条件と同等に扱えるようにする。次のセクションでは設定したペナルティ関数および最適化について述べる。

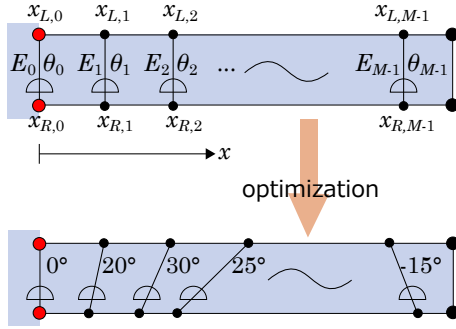


図 4.3: ストリップを分割する折り目と最適化計算後のイメージ

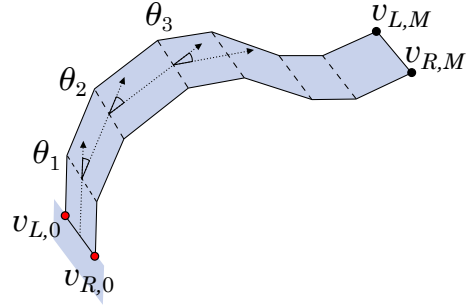


図 4.4: 3次元座標の計算

4.2 最適化による形状の修正

対象とする素材および得られる形状が左右対称であることから、ストリップの両端位置の制約は、「一方が対称面 (yz 平面) に乗る」という制約に置き換わる。3次元空間での各折り目の座標 $v_{L,i}$ 、 $v_{R,i}$ は、一方の端を固定して、折り目の位置と角度による座標変換を順に積算することで求まる (図 4.4)。この結果として、終端が対称面上に乗ることが制約条件となる。この条件付きの最適化問題は、ペナルティ法により解くものとし、解は Levenberg-Marquardt 法による反復計算で求める。提案システムで用いたペナルティ関数は次式で表され、この式を最小化するように折り目の配置が決定される。端点の拘束条件も同じ枠組みで実現するため、完全に条件を満たすことは保証されないが、ペナルティ関数の重みを大きくすることで実際には無視できるほどの誤差に納めることができる。

$$F_{\text{penalty}} = w_1 F_{\text{no-gap}} + w_2 F_{\text{smooth}} + w_3 F_{\text{fair}} + w_4 F_{\text{user-control}} \quad (1)$$

各項の説明は次のとおりである。

端点の位置制約 (図 4.6)

$$F_{\text{no-gap}} := (v_{L,M})_x^2 + (v_{R,M})_x^2 \quad (2)$$

終端の頂点を対称面上に乗せるための項。対称面を $x = 0$ とし、 x 座標値の二乗和とする。

滑らかさ (図 4.5)

$$F_{\text{smooth}} := \sum_i (-\theta_{i-1} + 2\theta_i - \theta_{i+1})^2 + F_{\text{end.angle}} \quad (3)$$

$$F_{\text{end.angle}} = ((v_{L,M-1} - v_{L,M}) \cdot \vec{v}_{p.o.s})^2$$

ストリップを滑らかに折れ曲がるようにするための項。折り目の角度の変化を前方差分と後方差分で計算し、これを与える。また端点の位置制約のみを維持すると、対称面上で

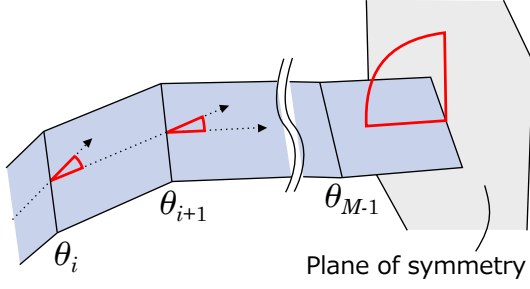


図 4.5: 滑らかさの計算

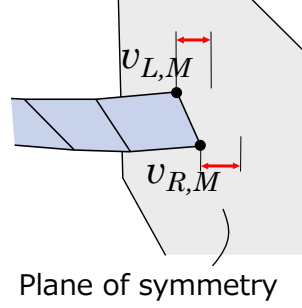


図 4.6: 端点の位置拘束

角度の付いた形状が生成されるため、これを防ぐためにストリップの終端が対称面に垂直に接するようにする項を加える。対称面を $x = 0$ としているため、 $\vec{v}_{plain_of_symmetry} = (1, 0, 0)$ である。

四角形領域の縮退排除 (図 4.7)

$$F_{fair} := \sum_i \left(\frac{1}{x_{L,i} - x_{L,i-1}} + \frac{1}{x_{R,i} - x_{R,i-1}} \right) \quad (4)$$

折り目が交わることを防ぐための項。隣り合う折り目との距離の逆数を与え、三角形に近い四角形が生じるのを防ぐ。

ユーザによる位置指定 (図 4.8)

$$F_{user_control} := \sum_i \left((v'_{L,i} - v_{L,i})^2 + (v'_{R,i} - v_{R,i})^2 \right) + dist(v_{L,M}, c_{symmetry})^2 + dist(v_{R,M}, c_{symmetry})^2 \quad (5)$$

ユーザが指定した頂点を指定位置に近づけるための項。3次元空間で頂点の目標座標を指定し、その座標との2乗距離を計算する。ある頂点に対応する目標位置を v' で表す。また対称面上の断面形状を表す曲線 $c_{symmetry}$ との2乗距離を与える。 $dist(v, c)$ は頂点 v から曲線までの最短距離である。ただし実装では、点と曲線の距離計算は反復計算を要するものであるため、実行速度の高速化のために細かい線分に区切ってその線分との距離を計算している。

提案システムでは、 $(w_1, w_2, w_3, w_4) = (0.1, 0.04, 0.01, 0.03)$ とした。これらの数値は経験的に得られたものである。 w_1 を大きくすることで可展制約を十分に満たす形状を得ることができる。初期状態では折り目は均等に配置し、角度は0度とした。 E_0 における折り角度は、脚部と平行な向きを0度として計算する。脚部の向きは地面 (xz 平面) と垂直としているが、4.4節では、これを变化させた例も示す。 E_0 に対応する頂点の3次元座標はユーザによって指定される。また E_n の位置および角度は E_0 から E_{n-1} までの折り目の結果として求まることから、計算が必要な変数の数 M は $M = (n-2) + (n-2) + (n-1)$ 個となる。したがって反復計算において解かれる行列は、大きさが $M \times M$ の疎行列となる。

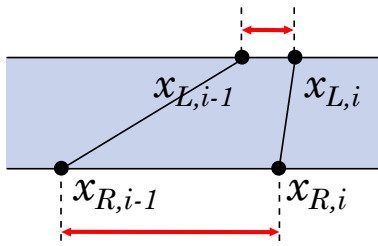


図 4.7: 四角形領域の縮退排除

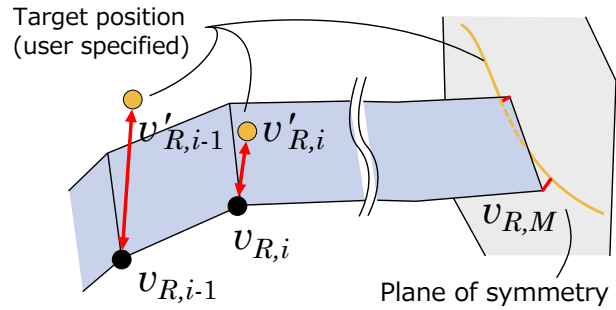


図 4.8: ユーザによる位置指定

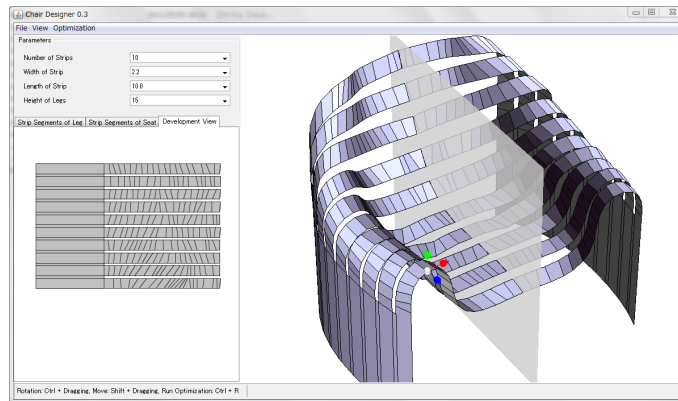


図 4.9: 提案システムの実行画面

4.3 ユーザ入力と作業プロセス

提案手法を実装し、図 4.9 に示す GUI を備えたシステムを開発した。ユーザははじめに展開図のサイズ、ストリップの数と幅などを指定する。その後、脚部の形状および対称面上の断面形状を入力すると、最適化により条件を満たす形状が自動生成されて大まかな形状を得ることができる（図 4.11）。それぞれの形状はベジェ曲線を用いて指定する。また、得られた結果に対して各頂点の目標位置を指定することが可能で、これによって意図した形状に近づける操作を行う（図 4.8）。一度指定した目標位置は保持されるため、複数の頂点に対して同時に位置指定をすることもできる。各頂点への目標位置の指定中は、選択しているストリップのみに最適化処理が実行されるようにした。最適化の計算は十分に高速であるため、マウスによる対話的な操作で、それに追従した形状変化を確認しながら編集作業を行うことができる。

さらに折り目の数を増やす細分割処理を実行できる。新しい折り目は、元の各折り線の間角度 0 度として配置される（図 4.10）。ただし折り目の数が増えると計算時間が増すため、最初に粗い折りで大まかな形状を決定し、その後実行することが望ましい。細分割と最適化処理を交互に適用することで、最終的に滑らかな形状を得ることができる。

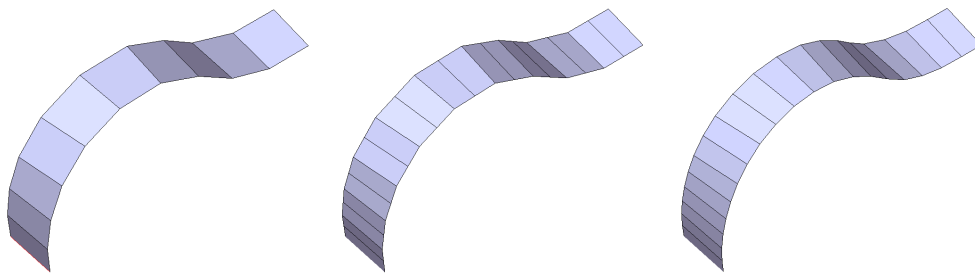


図 4.10: ストリップの細分割: 左から分割前、分割後、最適化後

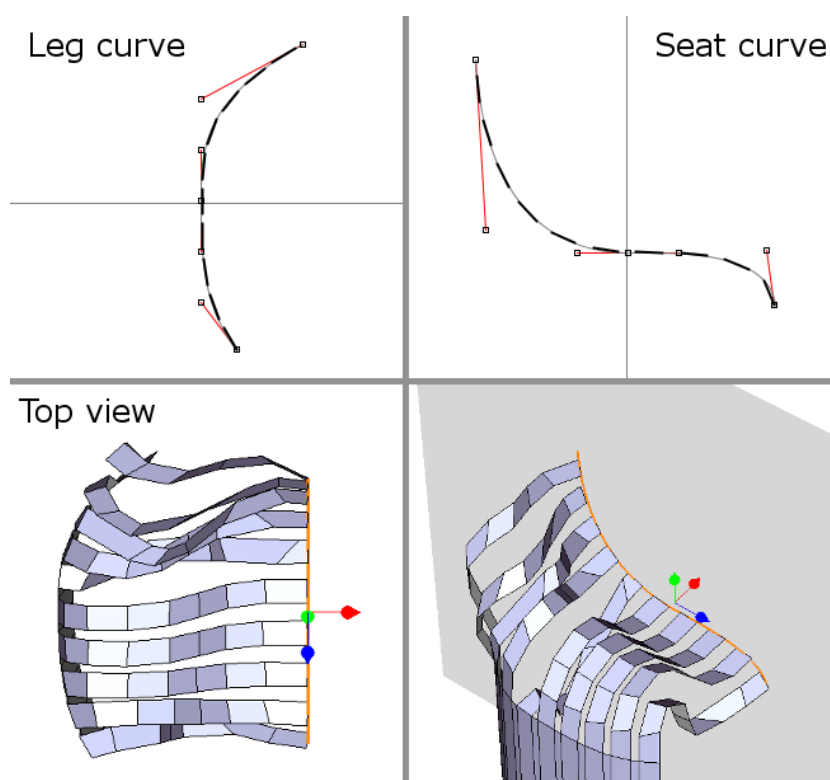


図 4.11: 脚部の形状と対称面上の断面形状の指定

4.4 結果と考察

提案システムを用いて生成した形状を図 4.12 に示す。実装は Java を用いて、描画に JOGL、行列演算に Matrix toolkits Java を使用し、Intel Xeon 2.66GHz 上で実行した。最適化を用いるアプローチでは、生成した形状が必ず展開できることは保証されない。提案手法では、対称面に終端がどれだけ近いかによって形状の正しさが評価できる。そこで、すべてのストリップのうち、終端の頂点と対称面との最大距離 G を展開図全体の長さ ($L2 + L3$) で割ることで誤差 $E = G / (L2 + L3)$ を計算した。その結果は、図 4.12 の例で最大 0.0015 であった。

作業中の最適化計算は、ストリップ 1 本あたり 100ms 未満でほぼリアルタイムで実行される。分割数を増やすと計算時間は増加するが、形状の変化が小さければ計算は早く収束する。そのため、より滑らかな形状が得られる 2 回の細分割 (33 個の折り目) の時点でも十分な速度が確保できる。しかし形状が大きく変形する最初の最適化や新しい面が作られる細分割の直後などでは実行速度が低下する。このような場合では、折り目の数が 9 のストリップ (これを初期状態とした) 1 本あたり 500ms ほどの時間を要した。計算が局所解に陥り適切に収束しない (いずれかのペナルティ関数が十分に小さくならない) 場合も発生し得るため、反復回数には制限を設けており計算を途中で打ち切るようにしている。収束しない場合でも、形状の編集では最適化処理が繰り返し実行されているため、ユーザ指定の制約条件を変化させることで容易に回避することができる。計算の打ち切りは実行速度を保つことにも貢献する。

図 4.12 の例題を生成するのに要した作業時間は 20 分程度であった。大まかな形を指定するのは容易だが、ストリップの波打や細かい位置修正は難しいことがわかった。例えば椅子の形であれば、座部を水平にしたいという要求がある。このような特定の部分に対する拘束条件やそれを指定するインタフェースがあれば、より効率的な作業が行えると思われる。図 4.12 の右端のモデルは、脚部の角度を地面と水平にしたものである。脚部の角度には 0 度を水平として 180 度までの任意の角度が考えられるが、自立できる形状であることを考慮し、角度の変更機能は実装しなかった。また、冒頭に記載した図 4.1 の椅子に似せた形状を作を試みたが、前述の理由により手作業での位置指定では大変な作業となった。そこで、紙で作られた図 4.1 のモデルを 3 次元計測器でスキャンし、得られた点群に近くなるように最適化を行った。

4.4.1 測定点群への近似

具体的な作業手順は次の通りである。

1. 測定した点群を単純化し、また孤立した面などを除去する。
2. ストリップ部分のみを個別のファイルに出力する。
3. それらのファイルをまとめて読み込み、ツール上で重ねて表示する (図 4.13)。
4. 大まかに手動で位置を合わせた後、最適化を行う。

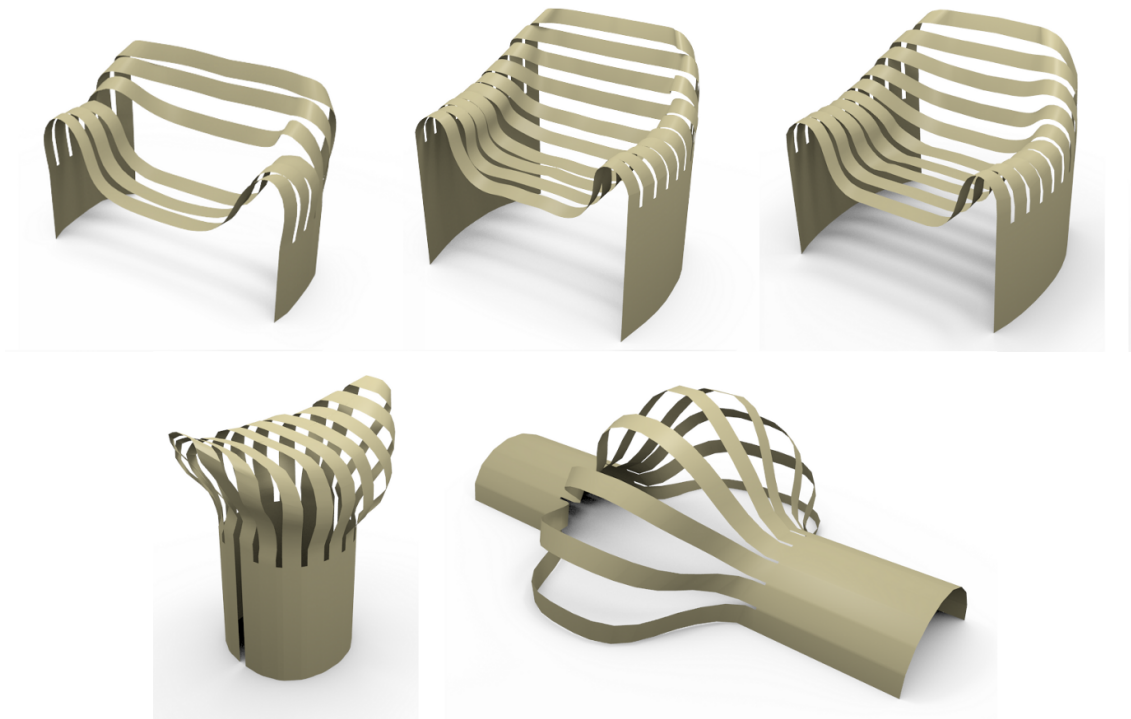


図 4.12: 作例: 左 2 つは 6 本と 10 本の椅子のモデル、中央はスキャンした点群に近づけたモデル、右 2 つはモニュメントのような形状

最適化は式 (1) に以下の項を加えて行った。

$$F_{prox} := \sum_i \left((nearest(v_{L,i}) - v_{L,i})^2 + (nearest(v_{R,i}) - v_{R,i})^2 \right) \quad (4.6)$$

ここで、 $nearest(v_i)$ は測定点群に含まれる頂点のうち v_i に最も近い頂点である。この項を加えることにより、より元の椅子に近いモデルを生成することができた。ただし、提案手法で作られるのはあくまで理想的なモデルである。実際の制作物には面のねじれが含まれていると考えられ、測定した点群にはさらに測定時の誤差も含まれる。そのため、点群と完全に一致させることは不可能である。

4.4.2 実制作

システム上で設計した形状が正しく形成できることを確認するため、2種類の素材を用いて実制作を行った。素材は、色画用紙と厚さ $1mm$ の塩化ビニル板である。また、計算された曲げ形状を正確に再現するため、素材の位置を固定するためのサポート部材を 3D プリンタで出力した。図 4.14 は、出力した型と、その表面に画用紙を貼り合わせたもの、および形成した塩化ビニル板である。塩化ビニル板の曲げ加工は、素材を加熱しながらサポート部材に押し

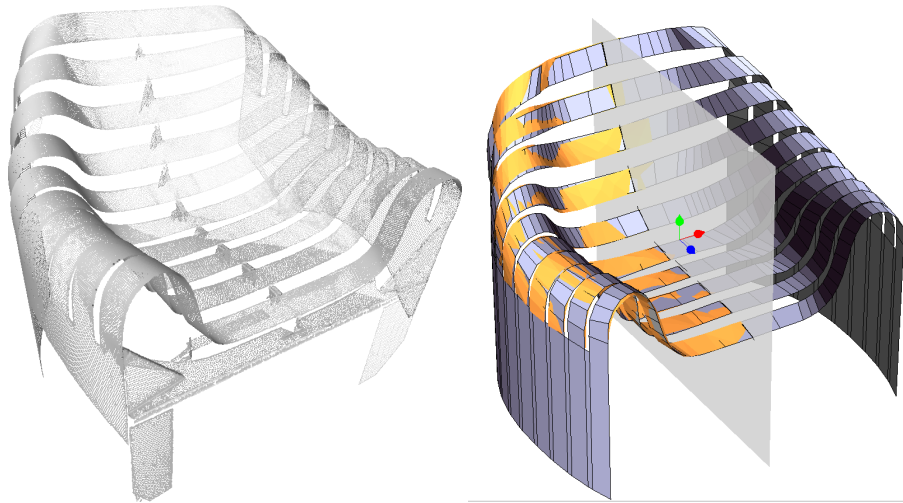


図 4.13: 3D スキャンで測定した点群データと編集集中の形を重ねて描画した図



図 4.14: 3D プリンタで出力したサポート部材とその表面に貼り合わせた画用紙、形成した塩化ビニル板; サイズ (W/H/D): $247 \times 233 \times 179\text{mm}$.

当てることで実現した。サポート部材として使用した 3D モデルは、システムで設計した形状を元にして閉じたソリッドモデルとして出力したものである。この際、スリットが垂直方向に重なっている部分など自動生成が難しい部分があったため、手動で修正を行った。

実際に制作した結果、出力した展開図は長辺方向がわずかに短く、サポート部材の表面を完全に覆えないことがわかった。この長さの不足は、素材の厚みによるものと推測される。サポート部材は設計した形状の内部を埋めるようにして作られているため、素材はその形状の外側に配置されることになり、ずれが生じる。より正確な形状を再現するためには、厚みを考慮した計算モデルが必要だと考えられる。

4.5 考察

本章では、スリットを入れた平板素材を折り曲げてできる形の設計について述べた。最適化計算によって折れ曲がる箇所の位置と角度を計算することで、展開可能な目的の形を達成している。提案したツールを用いることで、計算機上でこれまで実現することが困難だった形状を構築することが可能となる。

提案手法で扱うことができる展開図は大きく限定されたものである。よりバリエーションのある形を対象とするためには、スリットの位置や幅をずらすことが考えられる。本章で提案した幾何モデルで任意の多角形の素材とスリットを取り扱うためには、展開図の形とスリットの位置、そして折れ曲がる部分を指定するためのインタフェースが必要である。より大きな目標としては、切り込みを許容することで可能になる領域についての調査が挙げられる。目的の形状を実現するために必要な最低限の切り込みを推定することは、興味深いテーマである。

塩化ビニル板を用いた実制作においては、システム上での設計とほぼ同等の形状を実現することができた。ただし、素材の厚さの考慮が必要であることが確認された。素材の厚さの影響を最適化計算に組み込むと同時に、強度やバランスなどの解析ができれば、より実用的なシステムとして提供できると考えられる。

第5章 ユニット折り紙の組み合わせ構造の設計

第3章および第4章では、1枚の紙あるいは紙のような薄い素材による形の設計を行った。本章では複数の部品を組み合わせるユニット折り紙に対して、その基本構造となる多面体を設計するための手法について述べる。

ユニット折り紙は一枚の紙からユニットと呼ばれる小さな部品を折り出し、それを複数枚組み合わせることで、大きな構造を作り出す。一つのユニットはポケットと手と呼ばれるロック構造を持ち、一方の手を他方のポケットに差し込み、重ねて同時に折ることで2つのユニットを固定する。正多面体や準正多面体などの構造を組み上げることが多いため、1つの作品には単一の種類のユニットが用いられる。複数種類のユニットを用いる場合もあるが、それでもユニットの種類は数種類に留まる。

本章では、最も有名なユニットとして知られる園部式ユニットを対象として、そのユニットの組み合わせ構造で実現可能な形状の設計に取り組む。計算機上で対話的に構造の設計を行うことで、正多面体だけでなくより自由な立体形状の設計を可能にし、目的とする形状を組むのに必要なユニット数を事前に把握することができる。1つの園部式ユニットは組まれる多面体の辺に位置し、3つのユニットは正三角形（正三角錐）を成す。このため、園部式ユニットで組むことが可能な構造の設計は、正三角形集合の立体の設計問題に置き換えることができる。

この問題に対して、まず、5.1節で多面体の連結による対話的な設計手法を提案する^[80]。既知の正三角形集合の立体を連結してより大きな多面体を構築するという単純なアプローチで、立体の設計を試みた。結果として、ユーザは意図した形状がある程度作れる一方で、このアプローチでは正三角形集合で表現可能な立体の一部しか設計できないことが判明した。そして、設計できない形状の特徴や正三角形の面のみで構築可能な範囲が不明であるという大きな課題に直面した。

そこで、このような立体の性質を捉えるべく、面の接続関係と形状の関係についての調査に取り組んだ^[70]。ある決まった枚数の面で立体を構成する場合、面の接続関係が異なれば立体形状も異なる。どのような接続関係であれば可能であり、また不可能であるかを調べることで、正三角形の面で構築可能な立体の特徴が見出せると考えた。具体的には、三角形集合の多面体グラフを入力として、グラフから正三角形集合の多面体の構築を試みるという実験を行った。そして、連結操作では現れないような特徴的な構造や、正三角形の集合では表現できない構造を発見した。この実験結果は、5.2節に記載する。

対象とする立体形状には、既にいくつか知られている形があり、それらを簡単に紹介する。すべての面が等しい大きさの正三角形、つまり合同な正三角形でできた多面体は、数学的分

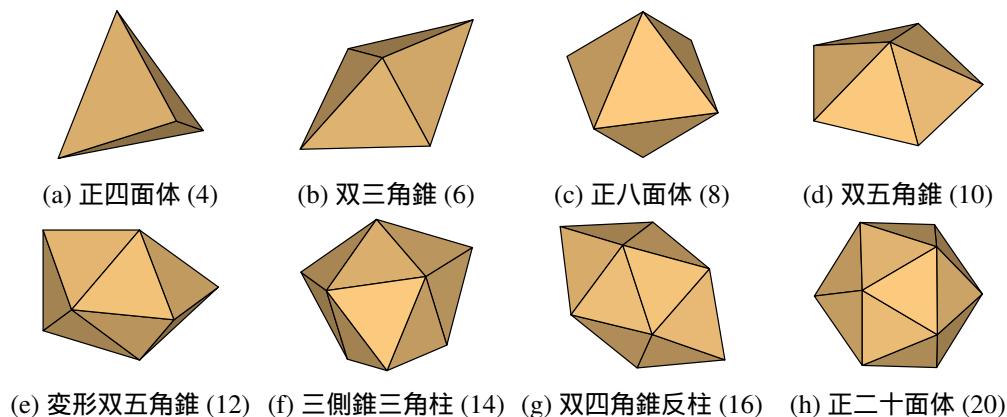


図 5.1: 凸デルタ多面体 (括弧内の数字は多面体を構成している面数)

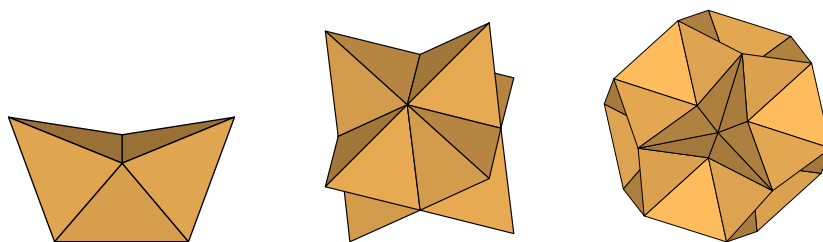


図 5.2: 凸でないデルタ多面体の例 (左:Boat、中:Stella octangula、右: Stellation of Icosahedron)

野ではデルタ多面体 (Deltahedron) と呼ばれている。その中でも、形が凸であるデルタ多面体は全部で図 5.1 に示す 8 種類のみであることが知られている^[76]。凸デルタ多面体のうち、面数が 4、8、20 の 3 種は正多面体であり、それぞれ正四面体、正八面体、正二十面体である。凸でないものの例としては図 5.2 に挙げたような形があり、非凸のデルタ多面体は無数に存在する。つまり、本研究ではこのような多面体を生成することとなる。その他、文献^[53]では頂点の形状が 2 種類のみデルタ多面体に着目し、バネモデル (立体の辺をバネとみなし、バネが自然長を保つように力を加えて変形する手法) によっても立体の生成ができることが報告されている。

5.1 対話的なデルタ多面体の設計システム

本節では、ユニット折り紙の組み合わせ構造であるデルタ多面体に対話的に設計するシステムについて述べる。既存のユニット折り紙を対象としたモデリング手法としては、田村らのシステムが挙げられる^[82]。これは、ユニットを模した簡易モデルと対話的なインタフェースによって仮想的にユニット折り紙が行えるものである。また、ボクセルと呼ばれる立方体集合を平行六面体に変形させることで、ボクセル集合からの正三角形集合を自動生成できる。しかしながら、1 つずつユニットを組み合わせるインタフェースは、どのように組み合わせれ

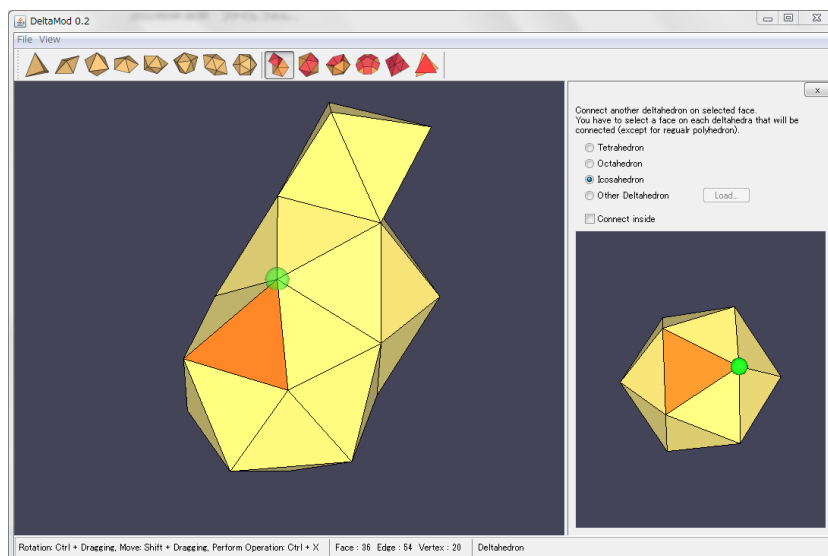


図 5.3: システム実行画面

ば目的の形状ができるかという知識をユーザが持っていなければならない。また、自動生成手法は大きなものが作れる一方で、ボクセル集合を一様に変形させるため、ユニットの組み合わせ方が非常に限定されているという問題がある。提案手法では、多面体の連結操作を基本とすることで、ユニットのつながり方を意識することなく目的とする形状を設計できるようにした。また、多面体を組み合わせるようにして作ること、自動変形では現れないユニットの組み合わせ方も内包している。

提案システムの実行画面を図 5.3 に示す。ユーザは初期状態としてデルタ多面体のうち 1 つを選択し、そこに編集操作を加えることで形状の設計を行う。本システムで可能な操作は、多面体の連結、引き伸ばし、錐形部位の凹凸反転、面の細分割の 4 つである。これらの操作は、図 5.1 に示したデルタ多面体を考察することで得たものである。特に、2 つ目の引き伸ばし操作は、既存ソフトウェアではいくつかの処理の組み合わせを要する難しい操作である。ユーザはマウスで目的の面を選択し、これらの編集操作を適用する。以下、操作の詳細を述べる。

5.1.1 多面体の連結

新しいデルタ多面体を作り出す単純な方法としては、すでにある多面体を連結することである。そこで、2 つのデルタ多面体を、面を合わせて連結する操作を実装した。正多面体をつなげる場合はどの面を合わせても同じ結果が得られるが、それ以外の場合では結果が異なるため、連結する面と向きを指定できる。編集時の形につなげたい形を読み込み、それぞれで連結する面と向きを指定して合わせることで、図 5.4 右のような結果を得ることができる。図 5.4 左は多面体を内側に連結する例である。ただし、内側への連結は面の突き抜けが発生する可能性がある。

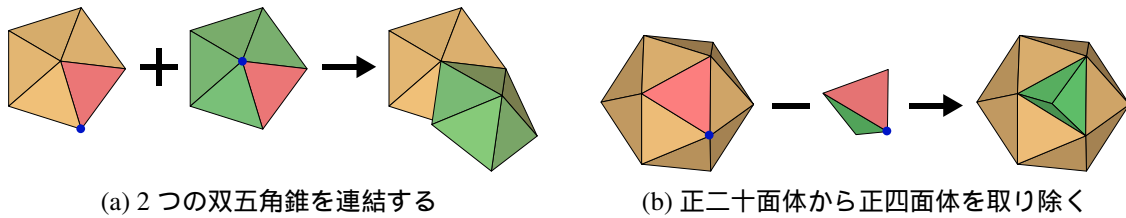


図 5.4: 多面体の連結操作

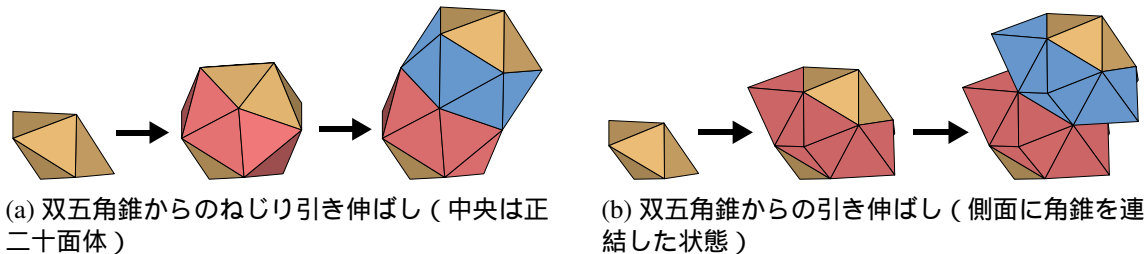


図 5.5: 引き伸ばし操作

これらの操作に加えて、更に正方形、正五角形の面に、それぞれ正四角錐、正五角錐を連結する操作も実装した。この操作により、辺の長さが等しいが四角形や五角形の面も含む多面体も初期入力として扱うことができる。

5.1.2 引き伸ばし

引き伸ばし操作は図 5.5 に示すような操作であり、真っ直ぐに伸ばすものとねじるように伸ばす2通りがある。この操作は、図 5.1 の双四角錐反柱の英名、“Gyroelongated square bipyramid”に由来する。まず、後者のねじり引き伸ばしから説明する。双五角錐と正二十面体に着目すると。双五角錐は、正二十面体の帯の部分（図 5.5 左）を取り除いて、上下の五角錐を連結した形であることがわかる。この帯の部分は側面が三角形の柱体であり、反角柱と呼ばれる。この反角柱の部分を重ねることによって新しい形を作ることでもあるため、この帯部分を引き伸ばす操作をシステムに組み込んだ。反角柱は双四角錐反柱にも存在し、双四角錐反柱から反角柱部分を取り除いたものが正八面体と一致する。前者の単純な引き伸ばし（図 5.5 右）は、反角柱でなく角柱（側面が四角形の柱体）を伸ばすことに相当する。この場合は、正四角形の面が発生してしまうため、それらの面を前述の連結操作で埋める必要がある。

5.1.3 錐形部位の凹凸を反転する

一部の凸デルタ多面体は、錐形の部分を持っている。この錐形の部分は、錐形の頂点を内側に押し込むことが可能である。この操作を正二十面体に適用した結果が図 5.6 である。5.1.1 節の多面体を内側に連結する操作はこの操作を包含しているが、この操作を行うためには、取

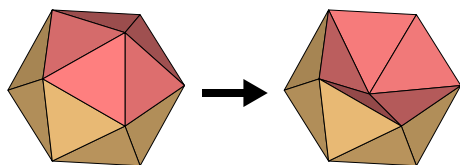


図 5.6: 正二十面体の錐形部位を反転する

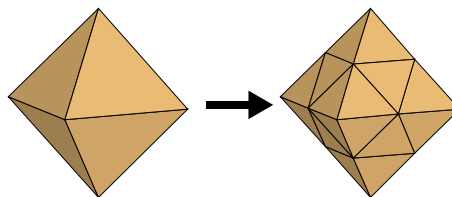
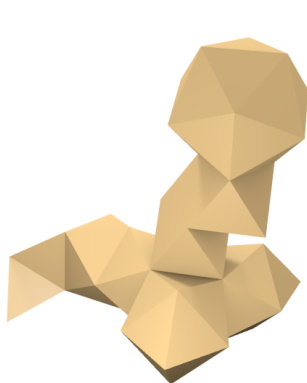
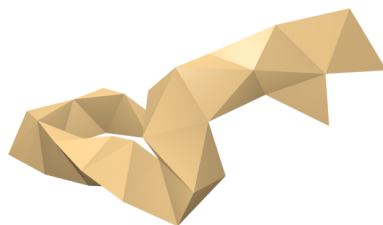


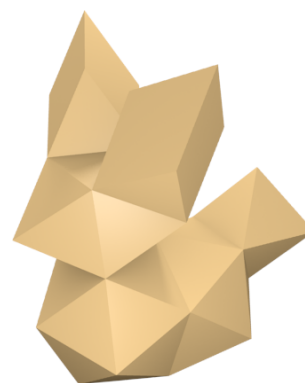
図 5.7: 正四面体の各面を 4 等分した例



(a) オットセイ (面数:90、辺数:135)



(b) ヘビ (面数:72、辺数:108)



(c) ウサギ (面数:64、辺数:96)

図 5.8: 対話的な編集によって作成したモデル例

り除く多面体を読み込み、面を指定することが必要であるため、操作を簡単にするという点では、この反転操作にも価値があると考えられる。

5.1.4 面の細分割

最後に面全体への操作として、各面を細分割する操作について述べる。これは、正三角形の隣り合う辺を n 等分した点をそれぞれ結び、その線分で区切られた n^2 個の小さい正三角形に、元の面を分割する操作である。全体の面数は n^2 倍になるが、細かい編集操作が可能となる。図 5.7 は正四面体の各辺を 2 等分（各面を 4 分割）する操作を示している。

5.1.5 結果と考察

前述の操作を組み合わせでデザインした立体を図 5.8 に示す。いずれも、作成に要した時間は数分程度である。複数の操作を組み合わせることにより、制約を意識することなく、ある程度目的とする形を作ることが可能であった。

図 5.8 のウサギのモデルを、図 5.9 に示す、折り紙で折った平行四辺形のユニットを用いて組み上げた。このユニットは菌部式ユニットと同様に、1 つが立体の 1 辺に相当し、3 枚で正

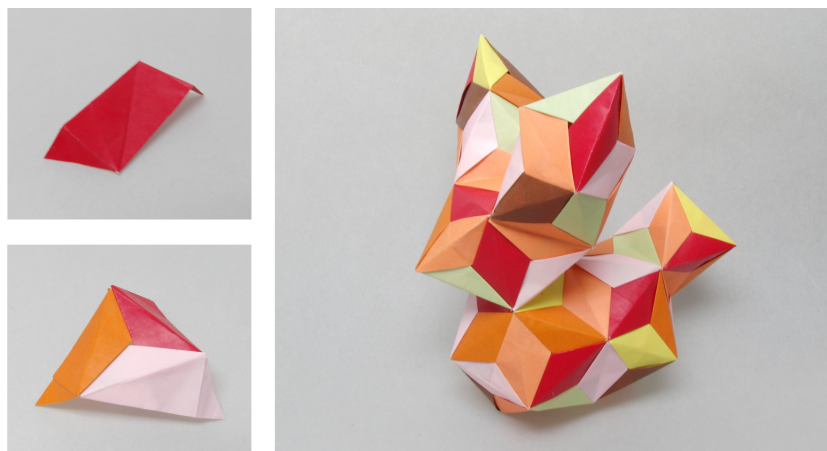


図 5.9: 折り紙ユニット 96 枚を組み合わせて制作したウサギのモデル (左上: 基本ユニット、左下: 3 枚組の正三角形)

三角形の面を成すように組み合わせることができる。ウサギのモデルには 96 本の辺が存在するため、96 個のユニットを使用した。本システムであらかじめモデリングしておくことにより、必要なユニットの数、配置などを十分に把握することができた。

提案システムで最も基本となる操作は連結操作であり、徐々に多面体を増やしていくというアプローチのため、膨らみをもつ形状が作りづらいという欠点がある。引き伸ばし操作はこの欠点をやや補うものであるが、それでも大きな構造を作ろうとすると細長くなってしまうことがわかった。

5.2 デルタ多面体のグラフと形状に関する調査

本節では、デルタ多面体の性質や可能な形状の調査のために行った、グラフからのデルタ多面体の構築実験について述べる。ここで、多面体の構造とは、頂点と辺からなるグラフ構造を指すものとする。デルタ多面体は面の形が限定されているため、このグラフ構造と形状が大きく関係している。そのため、グラフ構造を調べることで、形状の性質に関する知見が得られると考えた。

具体的には、三角形集合で表現されるすべてのグラフから、デルタ多面体として成立するものを抽出して、その構造について考察した。以下では、デルタ多面体グラフの定義と予備知識を説明した後、グラフ構造がデルタ多面体として成立するかどうかを判定するために、グラフの情報のみからデルタ多面体の構築を試みる手法について述べる。

5.2.1 デルタ多面体のグラフ

多面体の頂点と辺からなるグラフは、3 連結平面グラフである。これらのグラフは、三角形の面だけでなく多角形の面も含む。三角形の面のみを含むものは、3 次の 3 連結平面グラフ

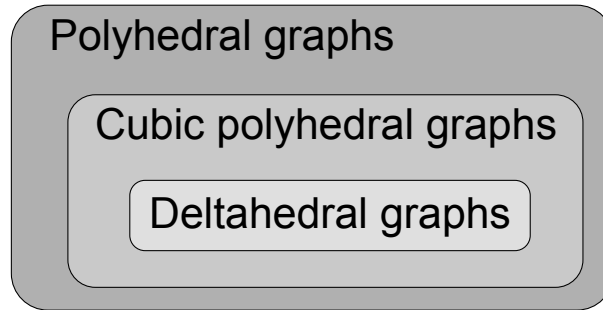


図 5.10: 多面体グラフとデルタ多面体グラフの包含関係

と呼ばれ、これは三角形の面のみで構成された多面体 (simple polyhedra) を構築できる。デルタ多面体は正三角形のみで構成された多面体であるため、simple polyhedra のサブクラスである。したがって、デルタ多面体のグラフも 3 次 3 連結平面グラフのサブクラスに属する。これらの包含関係は図 5.10 に示す。

本研究において、デルタ多面体グラフはデルタ多面体として構築可能なグラフとして定義する。デルタ多面体にも様々な種類が存在するが、ここでは自己交差がなく、二面角が 180 度でないもののみを許容する。例として、図 5.11 のすべての多面体は合同な正三角形のみできているが、中央の $(9_{24}, N)$ は面の交差を含んでおり、また右の $(8_{12}, N)$ は隣接した同一平面上の面、つまり 180 度の二面角を含んでいる。これら 2 つのような多面体は、デルタ多面体として取り扱わず、図 5.11 左の $(9_{17}, D)$ のような多面体のみをデルタ多面体とする。各図の下にあるコードは 2 つの数字と 1 つの文字で構成される。数字はそれぞれ、グラフに含まれる頂点数とグラフ固有のインデックスを示す。このインデックスは、既存のグラフ生成アルゴリズム^[9]の生成順序に従っている。文字はグラフがデルタ多面体を構築可能かどうかを識別するためのもので、‘D’は構築可能、つまりそのグラフがデルタ多面体グラフであり、‘N’は構築不可能、つまりデルタ多面体グラフではないことを示す。まとめると、例えば $(6_1, D)$ のグラフは、6 頂点のデルタ多面体グラフであり、グラフ生成アルゴリズムで一番最初に生成される。

デルタ多面体が構築できるかどうか焦点となるが、一つのグラフからは複数のデルタ多面体が構築できる可能性があるという点に注意が必要である。図 5.12 はデルタ多面体グラフとそのグラフから構築された 2 種類のデルタ多面体である。このように複数の異なる形状が存在し得るが、本研究では、少なくとも 1 つのデルタ多面体が構築可能であれば、そのグラフはデルタ多面体グラフであるとする。

5.2.2 グラフからの立体構築手法

前述のとおり、デルタ多面体は三角形集合の多面体に含まれるため、デルタ多面体グラフは最大でも 3 次 3 連結グラフの個数までしか存在しない。本手法は、面の隣接関係が異なるすべての 3 次 3 連結グラフを入力とする。グラフの列挙は広く研究されたテーマであり、様々

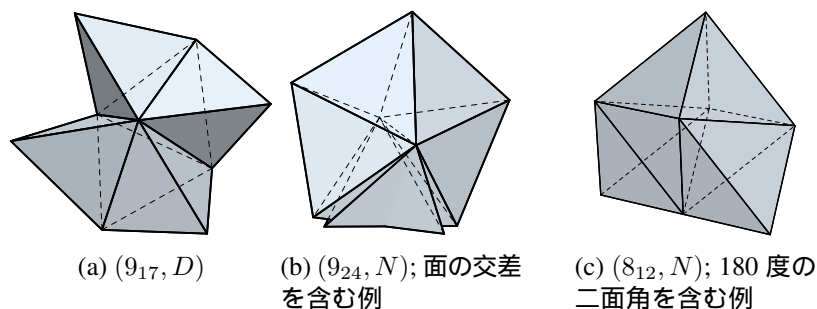


図 5.11: 正三角形集合の多面体; 上 1 つのみデルタ多面体であり、下 2 つはデルタ多面体ではない

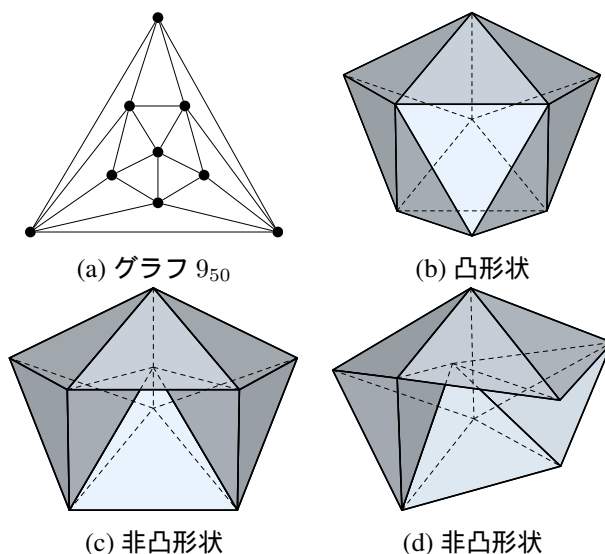


図 5.12: 9_{50} のグラフから構築できる複数のデルタ多面体

な列挙手法が存在するが、高速な列挙が可能な Brinkmann と McKay による plantri^[9] を使用した。

本手法は、グラフからデルタ多面体を構築することでそのグラフがデルタ多面体グラフかどうかを判定するが、その構築プロセスは 2 段階に分かれる。まず、入力されたグラフは直線の辺が交差しないように 2 次元平面に埋め込まれる。そして、その埋め込まれたグラフを持ち上げるようにして立体を構築し^[57]、反復計算によって少しずつ形状を変化させながらデルタ多面体に近づく。この時、辺の長さのみを考慮して変形するため、結果として得られる立体は面の交差などを含む可能性がある。最終的に得られた形状が、デルタ多面体の条件を満たす場合のみ、そのグラフがデルタ多面体グラフであるとする。

前述のとおり、非凸なものを含めるとデルタ多面体は無数に存在する。本稿では、頂点数を 10 以下に限定して実験を行った。頂点数 10 以下の 3 次 3 連結グラフは 306 個存在し、結

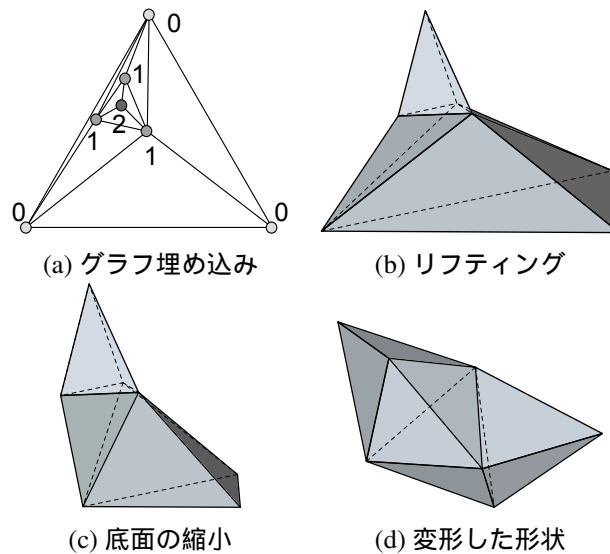


図 5.13: デルタ多面体の構築プロセス

果の傾向を確認するのに十分であると判断したためである。

グラフの埋め込み

平面グラフの埋め込みは様々な手法が提案されているが、ここではバネモデルを使用した Plestenjak のアルゴリズム^[54]を使用する。対象とするグラフのサイズは十分に小さいため、実時間で計算できる。このアルゴリズムは最初に 1 つ面を選択し、その面に含まれる頂点の位置を固定する。この面はランダムに選択し、それが 1 辺が単位長の正三角形をなすように配置するようにした。それ以外の頂点は、最初に選んだ面の内部に配置する。このとき、内部に配置する各頂点について periphericity p_v を計算する。Periphericity はグラフ理論における中心性に近いものであり、外側の多角形からの距離を示す。外側の正三角形に含まれる頂点の p_v は 0 であり、それに隣接する頂点の p_v は 1 となる。そして内側に向かうにつれて値は増加する。この periphericity は多面体構築プロセスにおいて、初期形状を生成するために使用される。図 5.13(a) はグラフ埋め込みの例を示す。埋め込みの結果は、最初に選ばれる面に依存する。

デルタ多面体の構築

最初に、正三角形ではない面を含む多面体を構築し、その多面体を変形させることでデルタ多面体を得る。この多面体は最終的には変形されるものの、なるべくデルタ多面体に近い方が望ましい。多面体はグラフの各頂点に高さを与えることで、ピラミッドのような形状を

構築する。各頂点の高さ h_v は p_v に比例して、次の式で求める。

$$h_v = Cp_v$$

ここで、 C はピラミッドの高さを決定する定数であり、 $C = 0.3$ とした。この値は経験的に得られたものであるが、最初に選択する面の一边の長さを 1 としたこととに依存する。このようにして得られた多面体は図 5.13(b) のようになり、さらに、辺の長さの差を小さくするため底面を縮小して、図 5.13(c) を得る。

そして、得られた多面体を最適化を用いて変形する。最適化は、辺の長さの差分を評価するペナルティ関数 F を最小化する。

$$F := \sum_i (L(E_i) - 1.0)^2$$

ここで、 $L(E_i)$ は辺 E_i の長さである。反復アルゴリズムは Levenberg-Marquardt アルゴリズムを使用し、線形方程式の解法は Gauss-Seidel 法を使用する。最初に構築するピラミッド型の形状によっては、計算が収束しない場合もあるが、その場合は、グラフの埋め込みをやり直すか、頂点の位置を手動で調整した。

最適化計算の収束は保証されていないが、面の交差を許容した場合に、すべての辺の長さを等しくするような頂点配置は必ず存在する。グラフの生成に必要な操作は次の 3 通りであり、最小の 4 頂点グラフは図 5.14 のように正四面体をなす。

(M_1) 次数 3 の頂点を追加する

(M_2) 辺を 1 つ取り除き、次数 4 の頂点を追加する

(M_3) 辺を 2 つ取り除き、次数 5 の頂点を追加する

これに対して 3 通りの操作を加えた後の形状を考えると、面の重なりを許容した場合、操作後の形状も正四面体の領域に収まる。操作 M_1 は面を 3 分割するように頂点を 1 つ追加する操作であるが、追加される面は分割元の面を底面とする正四面体のように実現できる。操作 M_2 によって追加される 2 つの面は、それぞれが分割前の面に重なるように配置できる。この際、それ以外の面は裏返されるが、結果として正四面体の形には収まる。操作 M_3 も同様に、全体の裏返しが必要になるが正四面体の形に収めるような配置が可能である。

5.2.3 結果

頂点数 10 以下のすべてのグラフに対して、デルタ多面体の構築可能性を調べた。本節では得られた結果と、構築された多面体の一部を示す。得られた多面体の例は付録にも記載してある。

表 5.1 は、デルタ多面体グラフとそれ以外のグラフの個数をまとめたもので、各頂点につき約 $2/3$ がデルタ多面体グラフで、残りはデルタ多面体を構築できないグラフであることが

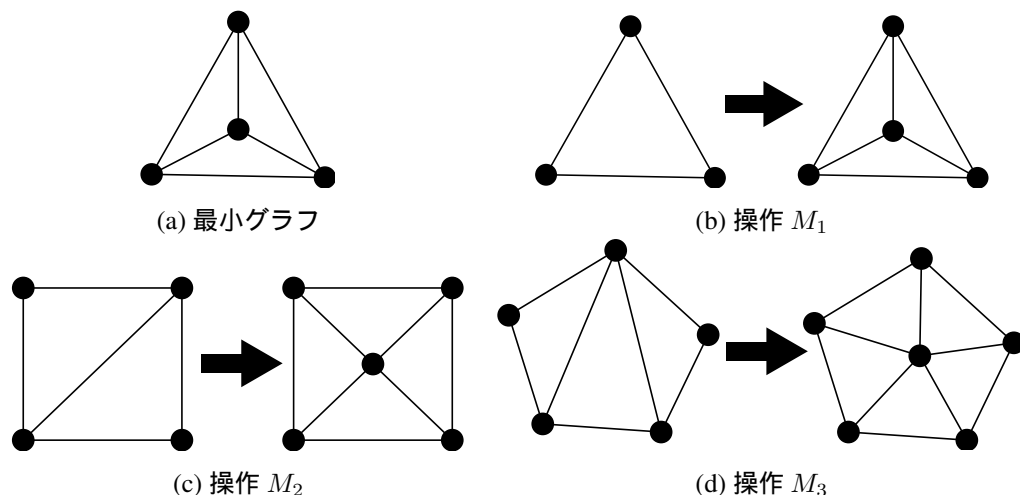


図 5.14: 三角形集合のグラフを生成するための 3 操作

表 5.1: デルタ多面体グラフの個数

頂点	4	5	6	7	8	9	10
三角形集合のグラフ	1	1	2	5	14	50	233
デルタ多面体グラフ	1	1	2	4	9	36	154
それ以外のグラフ	0	0	0	1	5	14	79

わかる。次節で詳細は述べるが、デルタ多面体を構築できないグラフが見つかった場合、頂点数を 1 つ増やしたグラフ群にも類似したグラフが現れる。したがってより頂点数の大きなグラフでも同じような傾向が見られると予想される。構築可能な割合の詳細は、調査範囲を広げて実験を行う必要がある。

図 5.15 と図 5.16 はそれぞれ、7 頂点と 8 頂点の全グラフから構築された多面体を示す。デルタ多面体グラフの項で述べたとおり、示した多面体は構築可能な一例であることに注意してほしい。ただし、頂点数 10 以下のグラフが取りうる形状の種類は多くとも 2、3 種類であったため、得られた形状を元に他の形状が存在するかどうかを視認で推測することは容易であった。判断が難しいケースについては、手動で頂点の位置を調整し、より良い初期多面体を与えながら調査した。

最適化により構築された多面体の辺は、すべて 1 に近くなる。誤差は残るものの、平均長と各辺の長さの最大誤差は 10^{-5} 以下であった。実験は 2.9 GHz Intel Core i7 CPU を搭載した計算機上で行い、ほとんどのグラフに対して数秒で計算できた。最適化に要する時間は、初期形状として与える多面体、つまり、グラフ埋め込みの結果に依存する。

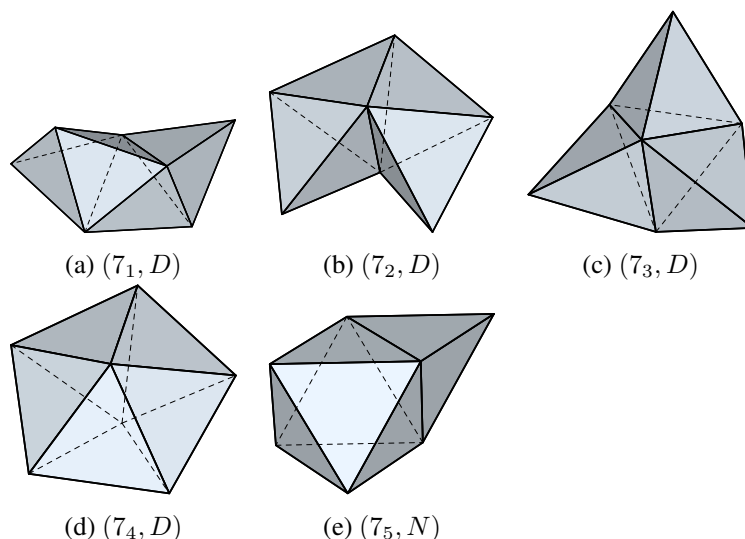


図 5.15: 7 頂点のグラフから構築された多面体

5.2.4 デルタ多面体として構築できないグラフ群の発見

ここでは、グラフ同士を比較することで、デルタ多面体として構築できない構造を検出する手法について述べる。実験結果からは、特定の構造をもつグラフは面の自己交差を起こすことがわかった。例えば、図 5.16 の 8_7 グラフは完全に平坦になってしまう。これと同じような形状はより頂点数の多いグラフにおいても出現するため、グラフの比較のみで判定できれば、頂点数の多いグラフ群に対して調査対象を絞り込むことが可能となる。

図 5.17 は同一の部分構造を含む異なるグラフと、対応する多面体を示している。2 つのデルタ多面体を面を合わせるようにして連結させる場合、それぞれの元の形状は変化することなく、また、連結される多面体のグラフは、もう一方のグラフの連結面の内側に埋め込まれる。この操作は部分グラフ同型問題を解くことで検出できる。したがって、構築できないグラフを 1 つ発見すれば、多面体を構築することなく、それと同種の構築できないグラフ群を得られることになる。

この手法により、 8_7 のグラフを、9 頂点と 10 頂点のすべてのグラフと比較した。その結果、図 5.18 に示す、デルタ多面体として構築できない 10 個のグラフを得た。部分グラフ同型問題は単純なバックトラック法^[71]を使用した。計算時間は 9 頂点のグラフ群に対して 140ms、10 頂点のグラフ群に対して 2500ms であった。

部分グラフ同型問題を解くことで検出できるのは、単一の面で連結する場合のみである。図 5.19 は単一の面による連結と、複数の面による連結を示した図であり、正四面体と正八面体を順番に連結している。2 つの操作の前後で元の形状は変化していないため同じ操作のように見えるが、中央から右への連結操作は、2 つの面を同時に連結させている。このため、左から中央への変形しか部分グラフ同型では検出できない。つまり、この検出手法において、右のグラフは 8_7 のグラフとは別種と判定される。なお、複数の面による連結は一般に元の形状

の変形を引き起こす操作であり、図 5.19 で形状が維持されているのは、これが正四面体と正八面体による空間充填の配置になっている例外的なケースのためである。

ところで、 7_5 のグラフは自己交差を含まない多面体として構築されるが、二面角の条件によりデルタ多面体ではない。このようなグラフは、部分グラフ同型問題の比較元のグラフにはならない。多面体を連結された後の形状では連結面周りの辺の二面角が変化し、条件を満たす可能性があるためである。

5.2.5 考察

本節では、デルタ多面体の性質調査を目的とした、グラフからデルタ多面体を構築する実験について述べた。実験は頂点数 10 以下のグラフを対象に行い、デルタ多面体を構築できない構造があることを確認した。また、部分グラフ同型問題を解くことで、不可能な構造を含むグラフ群を発見する手法を提案した。この枝刈り手法を用いて、より大きなサイズのグラフについて調査することが、今後の課題である。

現在のところ、デルタ多面体の性質の特定にはまだ至っていない。本研究で行ったグラフからデルタ多面体を構築することは、グラフの幾何的な構築問題^[20] (Graph realization と呼ばれる、グラフが立体を構築できるかどうかを判別する問題) と類似している。Graph realization における検出手法^[39, 59] を応用することで、より頑健な構築手法が実現できる可能性がある。

さらに、より大きな種数のグラフについても調査する必要がある。実験では種数 0 の球と同相なもののみを対象としたが、種数 1 以上のデルタ多面体も存在することが知られている。提案したデルタ多面体構築手法はグラフ埋め込みを使用するため、種数 0 にしか適用できない。そのため、種数に依存しない手法が望まれる。

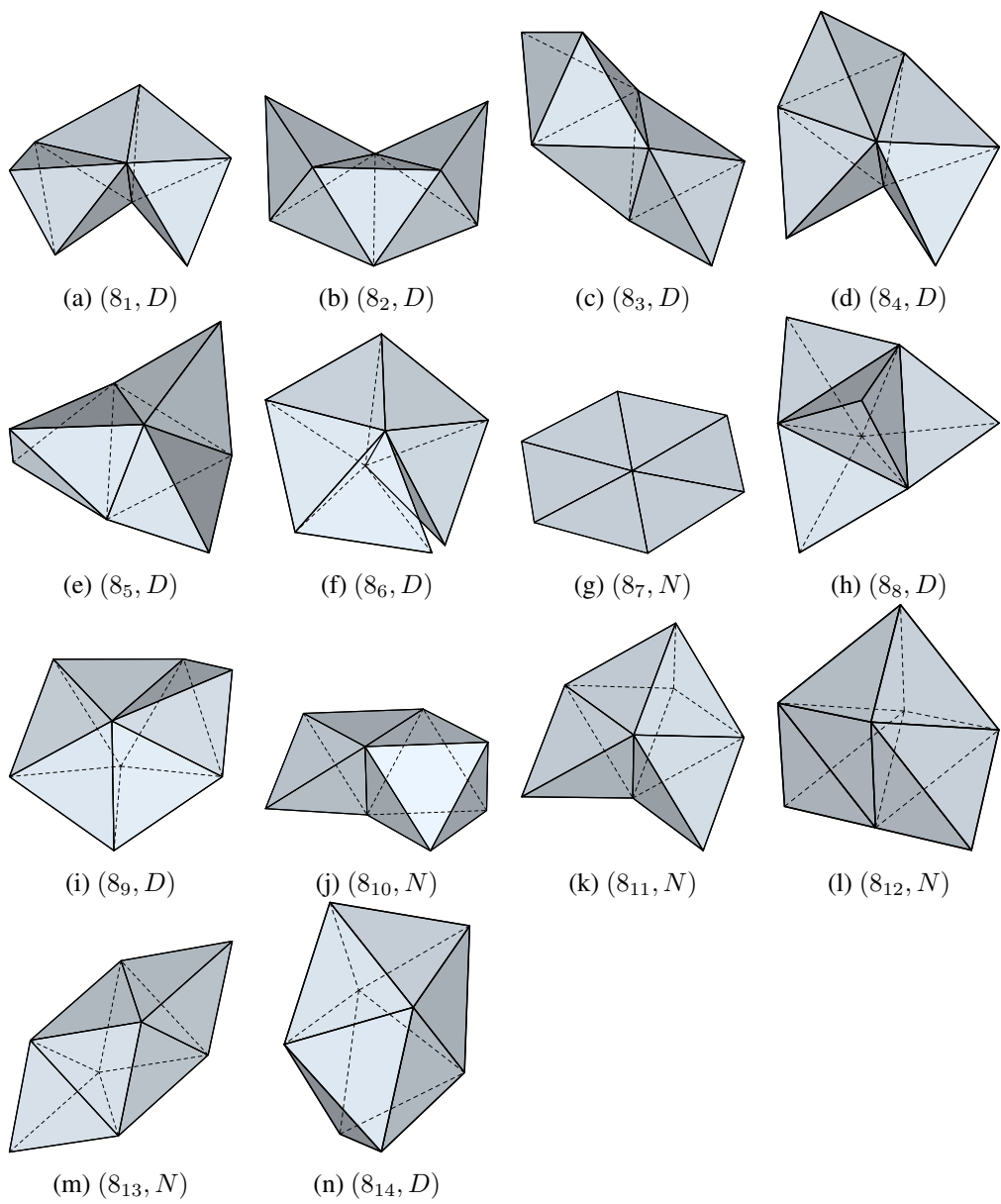


図 5.16: 8 頂点のグラフから構築された多面体

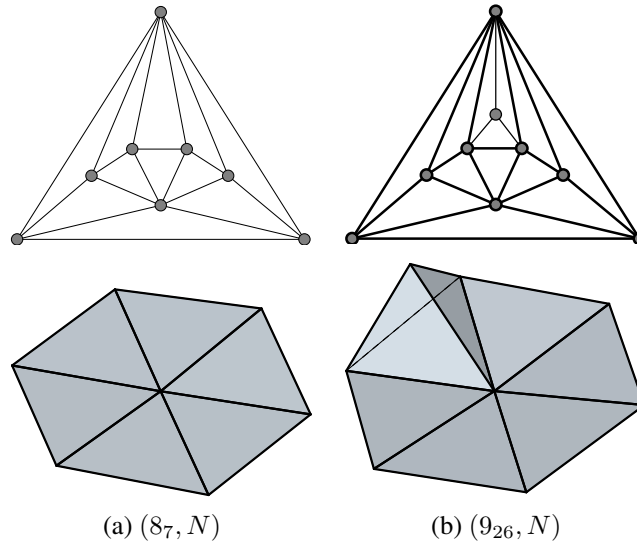


図 5.17: $(8_7, N)$ のグラフと正四面体を連結した $(9_{26}, N)$ のグラフ、および対応する立体

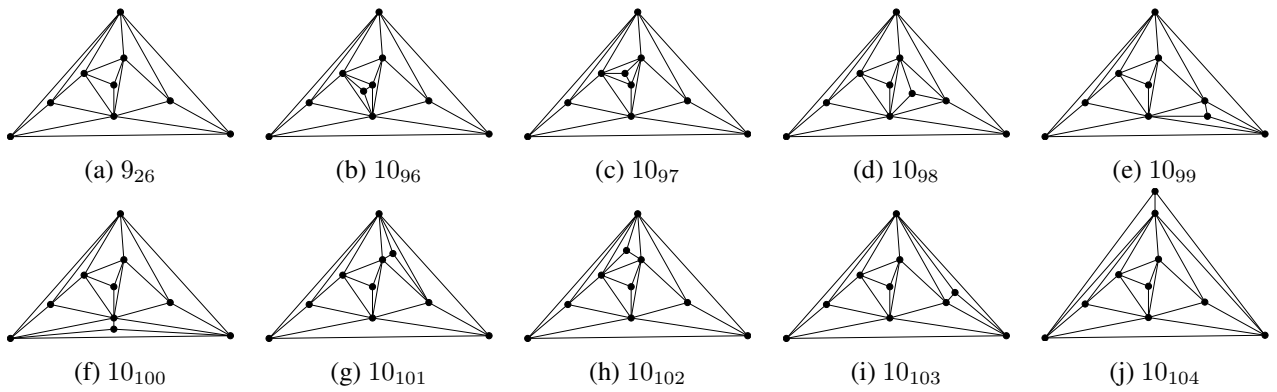


図 5.18: 8_7 のグラフを含むデルタ多面体を構築できないグラフ

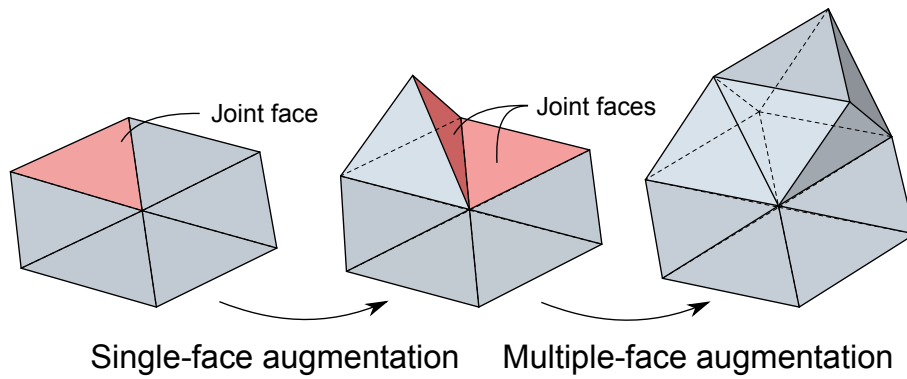


図 5.19: 単一面による連結と複数面による連結

第6章 結論と展望

6.1 結論

本研究では、計算機による折り紙の形状設計支援を目的として、形状提示による平坦折り紙作品の発見手法とスリットを含む平板素材の曲げ形状を設計する手法、そしてユニット折り紙の組み合わせ構造の設計手法を提案した。計算機なしでは実現できない折り畳み形状や組み合わせ方の列挙によって、これまで知られてこなかった新しい作品が発見できることを確認した。また、切り込みを許容することによる自由度の高い形状設計は、折り紙の枠を拡張した新しい可能性を切り開いた。

各テーマ毎の結論は以下に述べる。

6.1.1 折り畳み形状提示による作品発見

紙を折る回数を指定し、折り方をカドとカドを合わせるような折り操作のみに限定することで、可能な折り畳み形状をすべて列挙する手法を提案した。さらに、ユーザが入力した多角形との類似度を評価することで、列挙した形状の中から目的とする形状を抽出するシステムを実装した。また、カドを紙のフチに合わせるような折り方が一意に決まらない折り操作を含んだ、ランダムな折り畳み形状提示システムを提案した。提案手法によって計算機で自動生成された新しい作品を発見でき、また後者のシステムは、ウェブ上での公開と子供を対象としたワークショップを通して、幅広い人が作品発見を楽しめることを確認した。

6.1.2 スリットを含む平板素材の曲げ形状設計

平行なスリットを入れた長方形の素材の曲げ形状を設計する手法を提案した。スリットによって区切られた帯状領域を四辺形メッシュのストリップで表現し、全体として複数のストリップが結合した形状として定義した。そして、各頂点を対話的に編集すると同時に、最適化計算によって、素材の展開図がもつ拘束条件や曲げの滑らかさなどの制約を重み付けしたペナルティ関数を最小化する形状を求めることで、目的の形を実現した。出力形状は、塩化ビニル板を用いた実制作によって正しい形状が設計できていることを確認した。

6.1.3 ユニット折り紙の組み合わせ構造設計

ユニット折り紙の中でも特に代表的な菌部式ユニットの組み合わせ構造が、合同な正三角形集合の多面体と同じであることに着目し、そのような多面体を設計するための手法を提案した。合同な正三角形の集合で表現される多面体はデルタ多面体と呼ばれ、提案手法は、このデルタ多面体同士の連結や角柱部分の引き伸ばしなどの操作によって設計を行うものである。得られた構造を実際のユニットを使用して組み立てることで、ある程度自由な設計が可能であった。

また、デルタ多面体の構造についての調査も行った。凸形状のデルタ多面体は全部で8つ存在することが知られているが、凸でないものについては十分な研究がされていない。すべての面が三角形の多面体グラフに対して、あるグラフからデルタ多面体が構築可能かどうかを判別する実験を行った。結果として、デルタ多面体が構築できないようなグラフ構造が存在することを確認した。さらに、デルタ多面体を構築できないグラフを部分構造として含むグラフも、構築不可能であることを示した。

6.2 展望

前述の通り、平坦折り紙、立体的な曲げ形状、ユニット折り紙の組み合わせ構造を対象として、折り紙設計の新しい領域を開拓した。個別のテーマに残された課題や発展の方向は、各章の末尾に記載のため、本節では、共通する課題について言及する。

いずれのテーマにおいても、提案手法が対象とする形状には制限があるため、それらを補うように手法を拡張することが必要である。ただし、設計の自由度が高過ぎると人の手で探索する領域が大きくなってしまうため、汎用的であることが必ずしも良いとは限らない。そのため、対象を分割して個別の手法を提案していくことも重要となる。そうして、折り紙で可能な領域を少しずつ拡張し、価値ある形状を発掘していくことが折り紙設計の目標といえる。

では、その形状の価値はどのように評価すれば良いのだろうか。形状そのものの発見にも数学的な意義はあるが、序論で述べたような折り紙の工学応用に活用できれば、その形状や構造の価値はさらに高くなる。したがって、折り紙の世界に埋もれた形状に対して、どのようにして価値を見出し、発展させていくかが、折り紙という分野の大きな課題といえる。

折り紙の構造は工学的にも有用である一方、どのような構造がどの分野において活用できるかを判断することは容易ではない。折り紙の知識だけでなく、他分野の十分な知識を必要とするためである。複数の分野にまたがるエキスパートは極少数であり、そのために、応用の範囲が現状に留まっていると思われる。

この解決に向けて我々ができることは、本研究で達成したような新しい成果を幅広く公開し、従来では遊戯と考えられていた折り紙の認識を変えていくことである。また、専門的知識のない人が利用できるように、提案手法をソフトウェアとして提供することも重要といえる。様々な分野の開発者や研究者が折り紙の技術に触れる機会を増やせば、新たなアイデアの創発を促進できると思われる。そうして、我が国を代表とする文化である「折り紙」の技術を、日本のものづくり産業を活気づける要因とすることが強く望まれる。

謝辞

本研究を進めるにあたり、多くの方々のご協力とご助言をいただきました。

三谷純准教授には、これまでの研究成果を達成する鍵となるアイデアや方向性についての貴重なご指導をいただきました。私が折り紙の研究に取り組み始めたきっかけは、高専時代に拝読した三谷准教授の折り畳み形状推定の記事であり、これが今日に至る進路を決定した大きな要因となっています。研究のみならず、多くのアウトリーチ活動や雑誌の原稿執筆の機会も提供していただき、様々な経験を積むことができました。ここに深く感謝の意を表します。

金森由博助教には、論文を書く度に厳しい添削をしていただき、正確に記述するための技術をご指導をいただきました。また、常に学生の研究室で研究を進めてくださり、その姿から研究者や教育者としての生活を学ばせていただきました。深く感謝致します。

北陸先端科学技術大学院大学 上原隆平教授、本学 狩野均教授、志築文太郎准教授には、本論文の査読をしていただき、貴重なご意見とアドバイスをいただきました。ここに感謝の意を表します。

また、研究室の方々や先輩方には、研究生活を送る上で大変お世話になりました。飯塚里志氏は学部から6年間の研究生活を共に過ごし、特に、博士課程の3年間は同期の存在が支えとなりました。

最後に、学生生活を陰ながら支援していただいた家族に感謝致します。

2015年3月 鶴田直也

参考文献

- [1] Roger C. Alperin and Robert J. Lang. One-, two-, and multi-fold origami axioms. In Robert J. Lang, editor, *Origami 4: Fourth International Meeting of Origami Science, Mathematics, and Education*, pp. 371–393. A K Peters Ltd, 2009.
- [2] Antiprism. Polyhedron modelling software. <http://www.antiprism.com/>.
- [3] Zaha Hadid Architects. Venice architecture biennale - common ground - design - zaha hadid architects. <http://www.zaha-hadid.com/design/contribution-to-2012-venice-biennale-theme->
- [4] Esther M. Arkin, Michael A. Bender, Erik D. Demaine, Martin L. Demaine, Joseph S.B. Mitchell, Saurabh Sethia, and Steven S. Skiena. When can you fold a map? *Computational Geometry*, Vol. 29, No. 1, pp. 23 – 46, 2004. Special Issue on the 10th Fall Workshop on Computational Geometry, {SUNY} at Stony Brook.
- [5] Alex Bateman. Tess: origami tessellation software. <http://www.papermosaics.co.uk/software.html>.
- [6] Nadia M. Benbernou, Erik D. Demaine, Martin L. Demaine, and Aviv Ovadya. Universal hinge patterns for folding orthogonal shapes. In *Origami 5: Fifth International Meeting of Origami Science, Mathematics, and Education*, pp. 405–419. A K Peters/CRC Press, 2011.
- [7] Marshall Bern and Barry Hayes. The complexity of flat origami. In *Proceedings of the Seventh Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '96, pp. 175–183, Philadelphia, PA, USA, 1996. Society for Industrial and Applied Mathematics.
- [8] Pengbo Bo and Wenping Wang. Geodesic-controlled developable surfaces for modeling paper bending. *Comput. Graph. Forum*, Vol. 26, No. 3, pp. 365–374, 2007.
- [9] Gunnar Brinkmann and Brendan D. McKay. Fast generation of planar graphs. *MATCH Communications in Mathematical and in Computer Chemistry*, Vol. 58, pp. 323–357, 2007.
- [10] Oleksiy Busaryev, Tamal K. Dey, and Huamin Wang. Adaptive fracture simulation of multi-layered thin plates. *ACM Trans. Graph.*, Vol. 32, No. 4, pp. 52:1–52:6, July 2013.
- [11] Inc. Cardborigami. Cardborigami — instant space. <http://cardborigami.org/>, 2011.

- [12] Christoffer Cromvik and Kenneth Eriksson. Airbag folding based on origami mathematics. In *Origami 4: Fourth International Meeting of Origami Science, Mathematics, and Education*, 2009.
- [13] Erik D. Demaine, Martin L. Demaine, and Duks Koschitz. Reconstructing david huffman’s legacy in curved-crease folding. In Patsy Wang-Iverson, Robert J. Lang, and Mark YIM, editors, *Origami 5: Fifth International Meeting of Origami Science, Mathematics, and Education*, pp. 39–51. A K Peters/CRC Press, 2011.
- [14] Erik D. Demaine, Martin L. Demaine, and Jason S. Ku. Folding any orthogonal maze. In *Origami 5: Fifth International Meeting of Origami Science, Mathematics, and Education*, pp. 449–454. A K Peters/CRC Press, 2011.
- [15] Erik D. Demaine, Martin L. Demaine, and Joseph S. B. Mitchell. Folding flat silhouettes and wrapping polyhedral packages: New results in computational origami. In *Symposium on Computational Geometry’99*, pp. 105–114, 1999.
- [16] Erik D. Demaine and Joseph O’Rourke. *Geometric Folding Algorithms: Linkages, Origami, Polyhedra*. Cambridge University Press, 7 2007.
- [17] Chi-Wing Fu, Chi-Fu Lai, Ying He, and Daniel Cohen-Or. K-set tilable surfaces. *ACM Trans. Graph.*, Vol. 29, pp. 44:1–44:6, July 2010.
- [18] Flux Furniture. flux — join the fold. <http://www.fluxfurniture.com/>.
- [19] Mili Golan. Origametrica and the van hiele theory of teaching geometry. In *Origami 5: Fifth International Meeting of Origami Science, Mathematics, and Education*, 2011.
- [20] Branko Grünbaum. *Convex Polytopes*, Vol. 221 of *Graduate Texts in Mathematics*. Springer New York, 2 edition, 2003.
- [21] Kazuo Haga, Josefine C. Fonacier, and Masami Isoda. *Origamics: Mathematical Explorations Through Paper Folding*. World Scientific Pub Co Inc, 9 2008.
- [22] D. A. Huffman. Curvature and creases: A primer on paper. *IEEE Trans. Comput.*, Vol. 25, pp. 1010–1019, October 1976.
- [23] Akitaya Hugo, Ribeiro Matheus, Koike Carla, and Jose Ralha. Development of an intuitive algorithm for diagramming and 3d animated tutorial for folding crease patterns. In *Origami 5: Fifth International Meeting of Origami Science, Mathematics, and Education*, pp. 347–369. A K Peters/CRC Press, 2011.
- [24] Thomas Hull. On the mathematics of flat origamis. *Congressus Numerantium*, Vol. 100, pp. 215–224, 1994.

- [25] Thomas Hull. Counting mountain-valley assignments for flat folds. *Ars Combinatoria*, Vol. 67, p. 01845, 2003.
- [26] Humiaki Huzita. Axiomatic development of origami geometry. In H. Huzita, editor, *the First International Meeting of Origami Science and Technology*, pp. 143–159. Italy: Dipartimento di Fisica dell’Universita di Padova, 1991.
- [27] ISSEY MIYAKE INC. Brands : 132 5. issey miyake.
http://www.isseymiyake.com/en/brands/132_5.html.
- [28] Jacques Justin. Resolution par le pliage de l’equation du troisieme degre et applications geometriques. In H. Huzita, editor, *the First International Meeting of Origami Science and Technology*, pp. 251–261. Italy: Dipartimento di Fisica dell’Universita di Padova, 1991.
- [29] Toshikazu Kawasaki. On the relation between mountain-creases and valley-creases of a flat origami. In *Preceedings of the First International Meeting on Origami Science and Technology*, pp. 229–237, 1989.
- [30] Martin Kilian, Simon Flöry, Zhonggui Chen, Niloy J. Mitra, Alla Sheffer, and Helmut Pottmann. Curved folding. *ACM Trans. Graph.*, Vol. 27, pp. 75:1–75:9, August 2008.
- [31] Kaori Kuribayashi. *A Nobel Foldable Stent Graft*. PhD thesis, University of Oxford, 2004.
- [32] Robert J. Lang. Exhibitions. <http://www.langorigami.com/exhibitions/exhibitions.php>.
- [33] Robert J. Lang. *Origami Design Secrets: Mathematical Methods for an Ancient Art*. A K Peters Ltd, illustrated edition, 10 2003.
- [34] Robert J. Lang. Treemaker.
<http://www.langorigami.com/science/treemaker/treemaker5.php4>, 2006.
- [35] Robert J. Lang. Flat-unfoldability and woven origami tessellations. In *Origami 5: Fifth International Meeting of Origami Science, Mathematics, and Education*, pp. 475–488. A K Peters/CRC Press, 2011.
- [36] S.N. Le, Su-Jun Leow, Tuong-Vu Le-Nguyen, C. Ruiz, and Kok-Lim Low. Surface and contour-preserving origamic architecture paper pop-ups. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 20, No. 2, pp. 276–288, Feb 2014.
- [37] Simon X. Liao and Mirosław Pawlak. On image analysis by moments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 18, pp. 254–266, 1996.
- [38] Yang Liu, Helmut Pottmann, Johannes Wallner, Yong-Liang Yang, and Wenping Wang. Geometric modeling with conical meshes and developable surfaces. *ACM Trans. Graph.*, Vol. 25, pp. 681–689, July 2006.

- [39] Frank H. Lutz. Enumeration and random realization of triangulated surfaces. In Alexander I. Bobenko, John M. Sullivan, Peter Schröder, and Günter M. Ziegler, editors, *Discrete Differential Geometry*, Vol. 38 of *Oberwolfach Seminars*, pp. 235–253. Birkhäuser Basel, 2008.
- [40] Jun Maekawa and Kunihiko Kasahara. *Viva! Origami (In Japanese)*. Sanrio Publications, Tokyo, 1983.
- [41] Fady Massarwi, Craig Gotsman, and Gershon Elber. Paper-craft from 3d polygonal models using generalized cylinders. *Comput. Aided Geom. Des.*, Vol. 25, pp. 576–591, November 2008.
- [42] J. Mitani and H. Suzuki. Computer aided design for origamic architecture models with polygonal representation. In *Computer Graphics International, 2004. Proceedings*, pp. 93–99, June 2004.
- [43] Jun Mitani. Oriipa; origami pattern editor.
<http://mitani.cs.tsukuba.ac.jp/pukiwiki-oriipa/index.php>.
- [44] Jun Mitani. Recognition, modeling and rendering method for origami using 2d bar codes. In *Origami 4: Fourth International Meeting of Origami Science, Mathematics, and Education*, pp. 251–258, 2009.
- [45] Jun Mitani and Takeo Igarashi. Interactive design of planar curved folding by reflection. In *Pacific Conference on Computer Graphics and Applications - Short Papers*, pp. 77–81, Kaohsiung, Taiwan, September 21st - 23rd 2011.
- [46] Jun Mitani and Hiromasa Suzuki. Making papercraft toys from meshes using strip-based approximate unfolding. *ACM Trans. Graph.*, Vol. 23, pp. 259–263, August 2004.
- [47] Koryo Miura. Map fold a la miura style, its physical characteristics and application to the space science. In R Takaki, editor, *Research of Pattern Formation*, pp. 77–90. KTK Scientific Publishers, 1989.
- [48] Koryo Miura and Tomohiro Tachi. Synthesis of rigid-foldable cylindrical polyhedra. *International Society for the Interdisciplinary Study of Symmetry (ISIS-Symmetry)*, Vol. Special Issues for the Festival-Congress Gmuend, pp. 204–213, August 23-28 2010.
- [49] Shinya Miyazaki, Takami Yasuda, Shigeki Yokoi, and Jun ichiro Toriwaki. An origami playing simulator in the virtual space. *Journal of Visualization and Computer Animation*, Vol. 7, No. 1, pp. 25–42, 1996.
- [50] Meenakshi Mukerji. *Marvelous Modular Origami*. A K Peters/CRC Press, illustrated edition edition, 4 2007.

- [51] Rahul Narain, Tobias Pfaff, and James F. O’Brien. Folding and crumpling adaptive sheets. *ACM Trans. Graph.*, Vol. 32, No. 4, pp. 51:1–51:8, July 2013.
- [52] Yuri Naruse and Jun Inokuma. Naruse inokuma architects, <http://www.narukuma.com/main.html>.
- [53] George Olshevsky. Breaking cundy’s deltahedra record. [http://www.interocitors.com/polyhedra/papers/Polytopics 28 - Breaking Cundy’s Deltahedra Record.pdf](http://www.interocitors.com/polyhedra/papers/Polytopics%20-%20Breaking%20Cundy's%20Deltahedra%20Record.pdf).
- [54] Bor Plestenjak. An algorithm for drawing planar graphs. *Softw. Pract. Exper.*, Vol. 29, No. 11, pp. 973–984, September 1999.
- [55] Helmut Pottmann, Alexander Schiftner, Pengbo Bo, Heinz Schmiedhofer, Wenping Wang, Niccolo Baldassini, and Johannes Wallner. Freeform surfaces from single curved panels. *ACM Trans. Graph.*, Vol. 27, pp. 76:1–76:10, August 2008.
- [56] Kenneth Rose, Alla Sheffer, Jamie Wither, Marie-Paule Cani, and Boris Thibert. Developable surfaces from arbitrary sketched boundaries. In *Proceedings of the fifth Eurographics symposium on Geometry processing*, pp. 163–172, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association.
- [57] Günter Rote. Realizing planar graphs as convex polytopes. In *Proceedings of the 19th International Conference on Graph Drawing, GD’11*, pp. 238–241, Berlin, Heidelberg, 2012. Springer-Verlag.
- [58] RVTR. resonant chamber — rvtr. <http://www.rvtr.com/research/resonant-chamber/>, 2012.
- [59] Lars Schewe. Nonrealizable minimal vertex triangulations of surfaces: Showing nonrealizability using oriented matroids and satisfiability solvers. *Discrete Comput. Geom.*, Vol. 43, No. 2, pp. 289–302, January 2010.
- [60] Hiroshi Shimanuki, Jien Kato, and Toyohide Watanabe. Constituting origami models from sketches. In *Proceedings of the Pattern Recognition, 17th International Conference on (ICPR’04) Volume 1 - Volume 01*, ICPR ’04, pp. 628–631, Washington, DC, USA, 2004. IEEE Computer Society.
- [61] Mayank Singh and Scott Schaefer. Triangle surfaces with discrete equivalence classes. *ACM Trans. Graph.*, Vol. 29, pp. 46:1–46:7, July 2010.
- [62] John Smith. Some thoughts on minimal folding. In *British Origami Society Magazine*, No. 162, 1993.

- [63] Justin Solomon, Etienne Vouga, Max Wardetzky, and Eitan Grinspun. Flexible developable surfaces. *Comp. Graph. Forum*, Vol. 31, No. 5, pp. 1567–1576, August 2012.
- [64] Tomohiro Tachi. Simulation of rigid origami. In *Origami 4: Fourth International Meeting of Origami Science, Mathematics, and Education*, pp. 175–187. A K Peters, 2009.
- [65] Tomohiro Tachi. Freeform rigid-foldable structure using bidirectionally flat-foldable planar quadrilateral mesh. In *Advances in Architectural Geometry 2010*, pp. 87–102, 2010.
- [66] Tomohiro Tachi. Origamizing polyhedral surfaces. *IEEE Transactions on Visualization and Computer Graphics*, Vol. 16, pp. 298–311, March 2010.
- [67] Naoya Tsuruta, Mitani Jun, Kanamori Yoshihiro, and Fukui Yukio. A cad system for diagramming origami with prediction of folding processes. In *Origami 5: Fifth International Meeting of Origami Science, Mathematics, and Education*, pp. 335–345. A K Peters/CRC Press, 2011.
- [68] Naoya Tsuruta, Jun Mitani, Yoshihiro Kanamori, and Yukio Fukui. An exploring tool for simple flat origami based on random foldings. In *International Conference on Simulation Technology (JSST2013)*, 2013.
- [69] Naoya Tsuruta, Jun Mitani, Yoshihiro Kanamori, and Yukio Fukui. Interactive design of 3d geometry made by bending inextensible sheet material containing slits. *International Journal of CAD/CAM*, Vol. 13, No. 2, pp. 23–29, 2013.
- [70] Naoya Tsuruta, Jun Mitani, Yoshihiro Kanamori, and Yukio Fukui. Enumeration of deltahedral graphs with up to 10 vertices. In Hans-Peter Schröcker and Manfred Husty, editors, *Proceeding of the 16th International Conference on Geometry and Graphics (ICGG2014)*, pp. 328–335, Innsbruck, Austria, 2014. insbruck university press.
- [71] J. R. Ullmann. An algorithm for subgraph isomorphism. *J. ACM*, Vol. 23, No. 1, pp. 31–42, January 1976.
- [72] Takuya Umesato, Toshiki Saitoh, Ryuhei Uehara, Hiro Ito, and Yoshio Okamoto. The complexity of the stamp folding problem. *Theoretical Computer Science*, Vol. 497, No. 0, pp. 13 – 19, 2013. Combinatorial Algorithms and Applications.
- [73] Paula Versnick. Orihouse. <http://home.tiscali.nl/gerard.paula/origami/orihouse.html>.
- [74] Patsy Wang-Iverson, Robert J. Lang, and Mark Yim, editors. *Origami 5: Fourth International Meeting of Origami Science, Mathematics, and Education*. A K Peters Ltd, 7 2011.
- [75] Robert Webb. Stella: Polyhedron navigator. <http://www.software3d.com/Stella.php>.
- [76] Eric W. Weisstein. “deltahedron.” from mathworld—a wolfram web resource. <http://mathworld.wolfram.com/Deltahedron.html>.

- [77] Lifeng Zhu, Takeo Igarashi, and Jun Mitani. Soft folding. *Computer Graphics Forum*, Vol. 32, No. 7, pp. 167–176, 2013.
- [78] 古田陽介, 木本晴夫, 三谷純, 福井幸男. マウスによる仮想折り紙の対話的操作のための計算モデルとインタフェース (コンピュータグラフィックス, 特集 インタラクションの理解とデザイン). 情報処理学会論文誌, Vol. 48, No. 12, pp. 3658–3669, dec 2007.
- [79] 新宮文明. おりがみしようよ! - めだまシールつき (実用 BEST BOOKS). 日本文芸社, 12 2005.
- [80] 鶴田直也, 三谷純, 金森由博, 福井幸男. 正三角形を基本単位とする立体の生成システム. 日本図学会春季大会 学術講演論文集, pp. 15–20, 2011.
- [81] 鶴田直也, 三谷純, 金森由博, 福井幸男. 形状列挙に基づく幼児向け折り紙作品の創作支援システム. 折り紙の科学, Vol. 2, No. 1, pp. 33–44, 2012.
- [82] 田村友和, 高井昌彰, 高井那美. ユニット折り紙を用いた 3 次元メッシュモデルの近似形状構築. 研究報告グラフィックスと CAD (CG), 第 2010-CG-141 巻, pp. 1–6, 2010.

付 録 A 自動生成された折り紙形状



図 A.1: 1 回折り (全 4 通り)

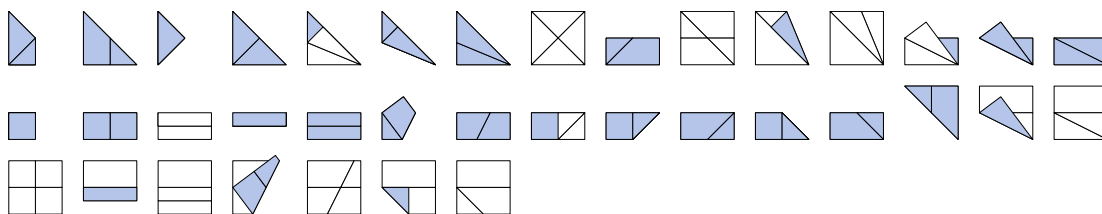


図 A.2: 2 回折り (全 37 通り)

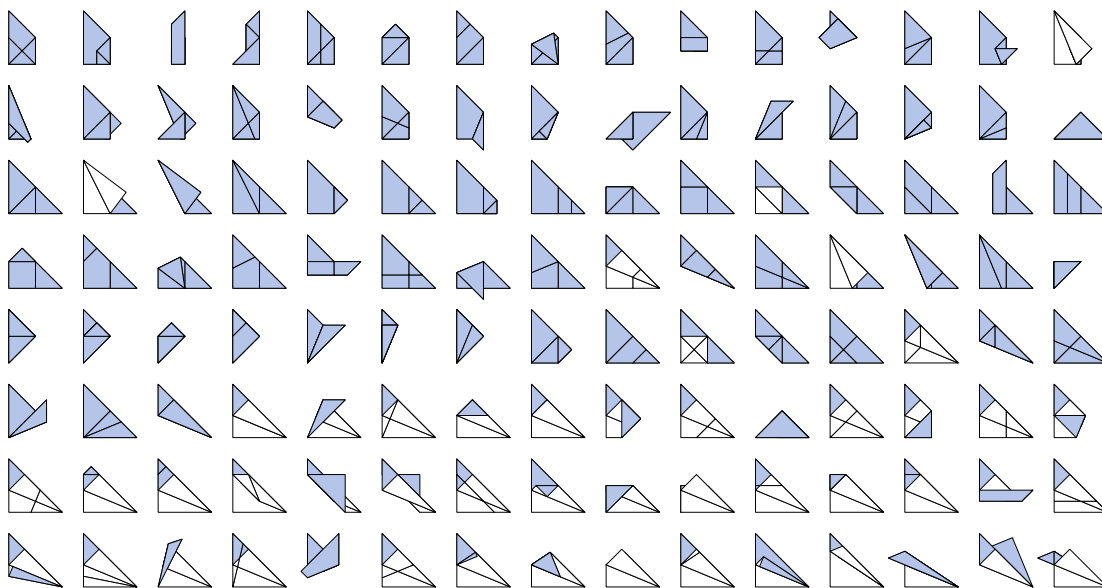


図 A.3: 3 回折り (1,507 通り中の 120 例)

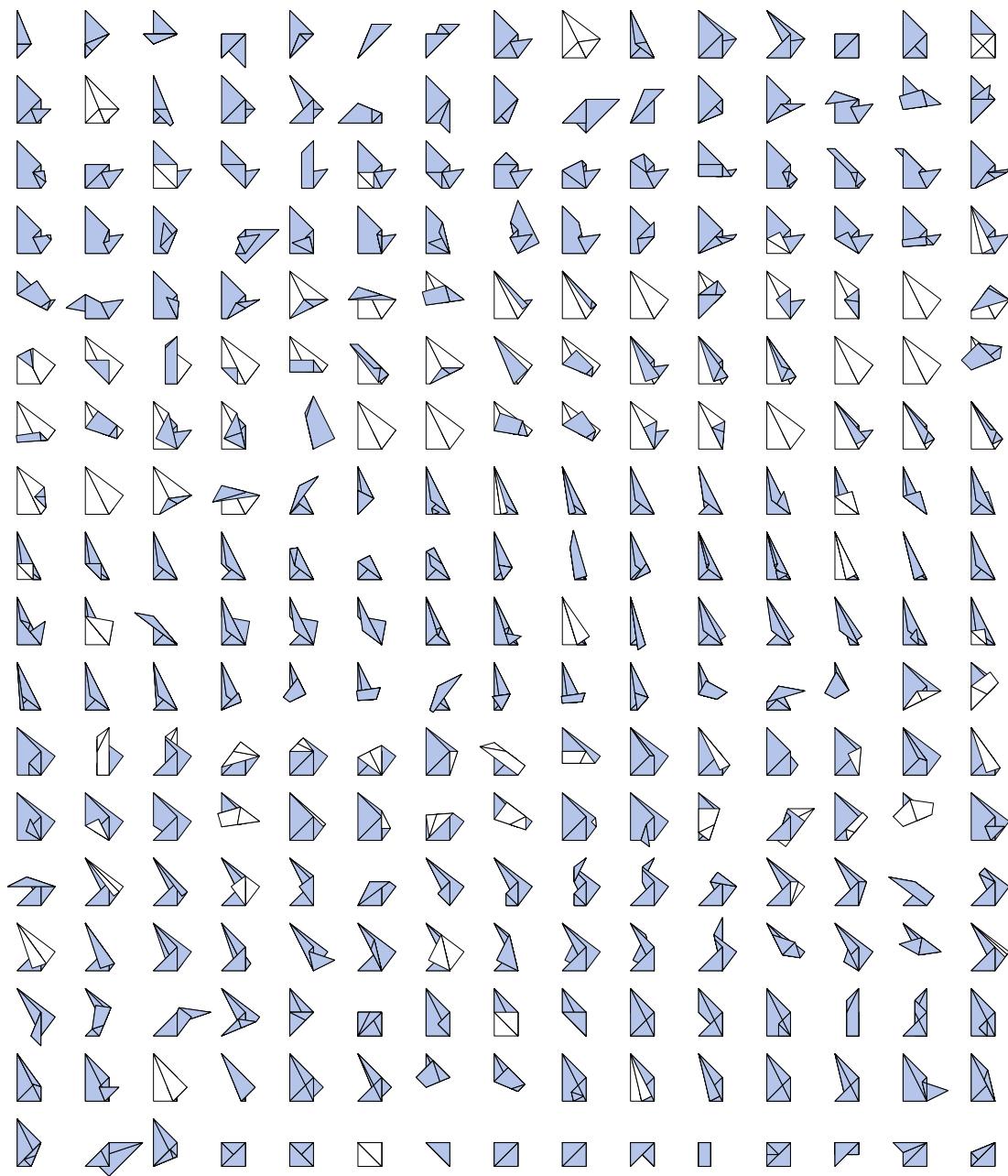


図 A.4: 4 回折り (136,284 通り中の 270 例)

付録B 3次3連結平面グラフを各面が正三角形になるよう構築した多面体

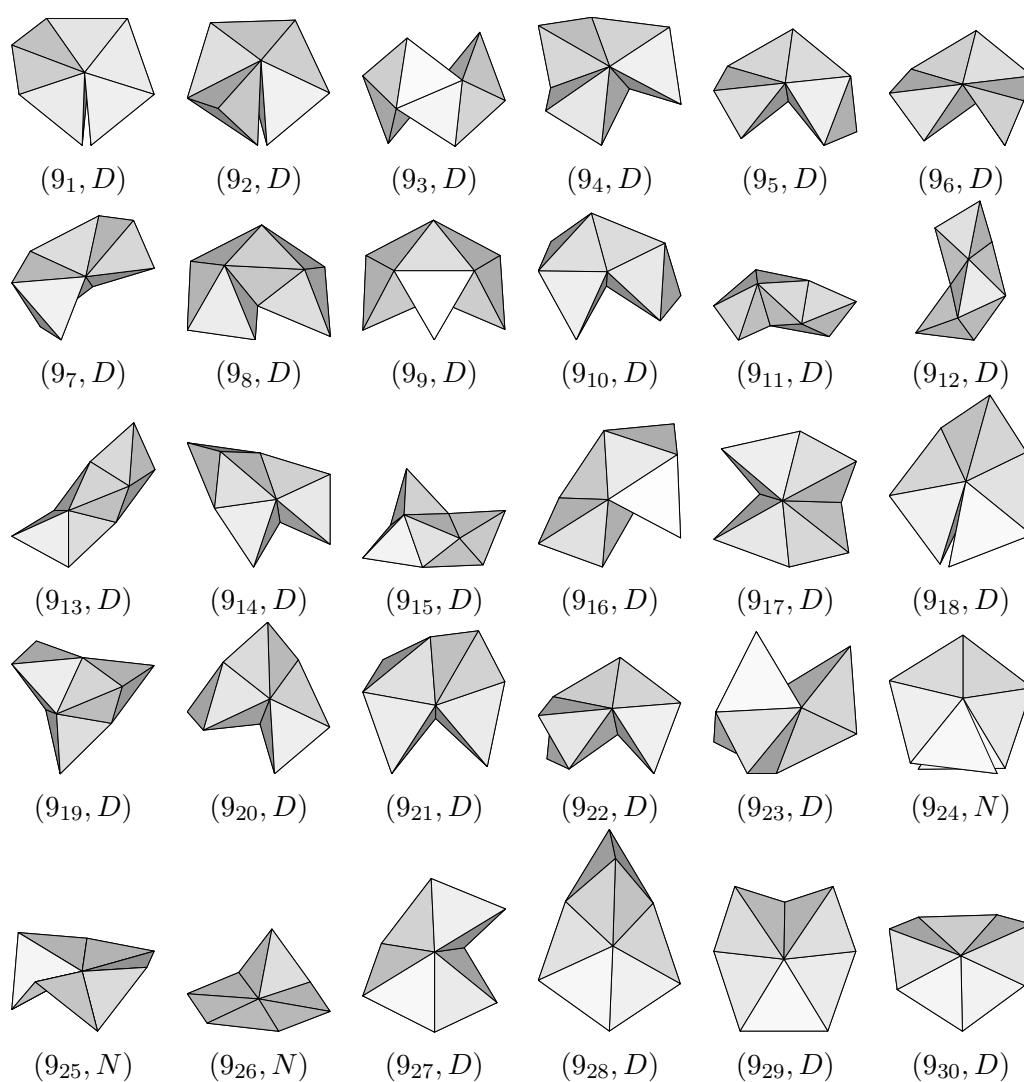


図 B.1: 9 頂点のグラフから構築した多面体 (50 通り中の 30 例)

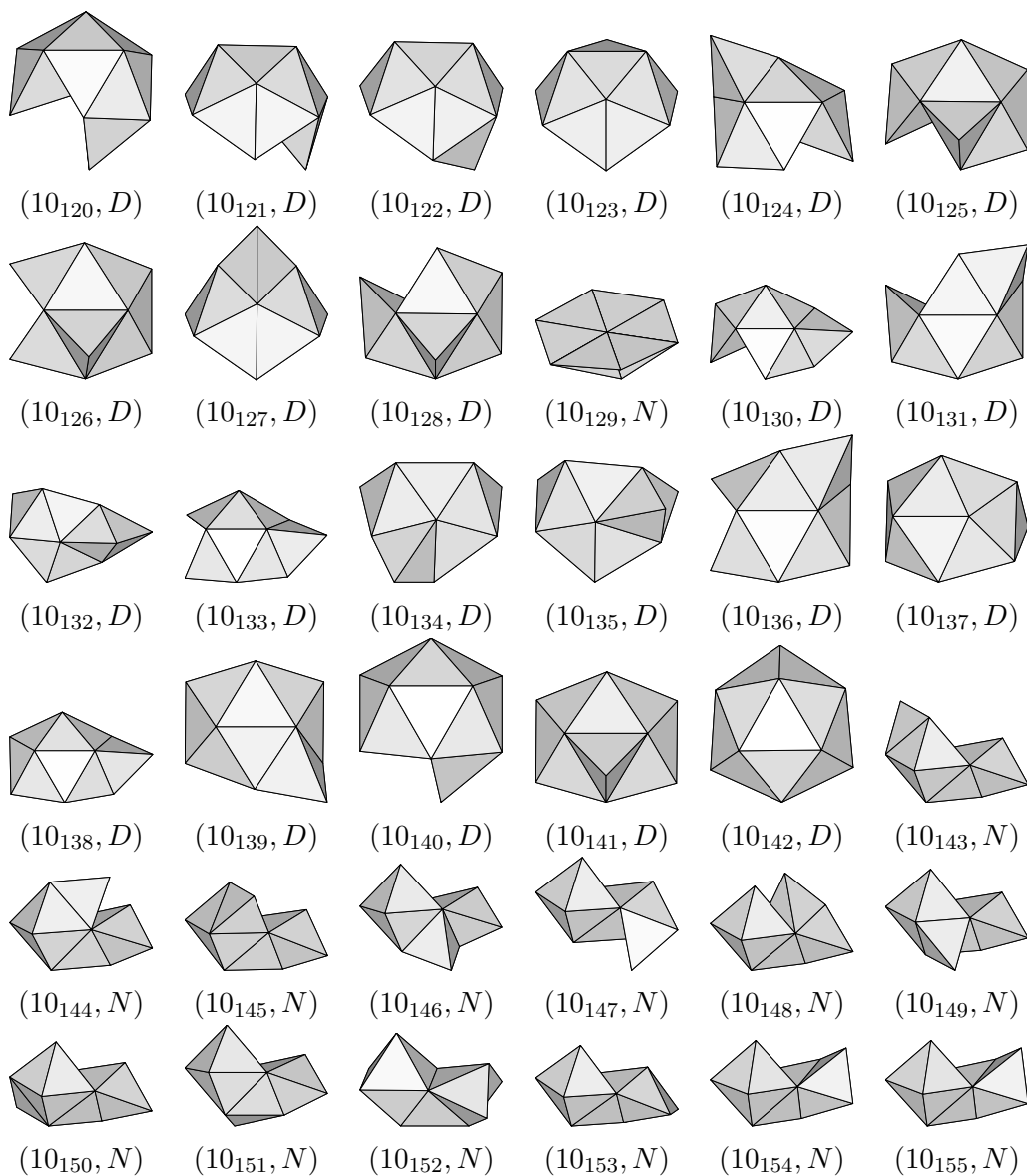


図 B.2: 10 頂点のグラフから構築した多面体 (233 通り中の 36 例)